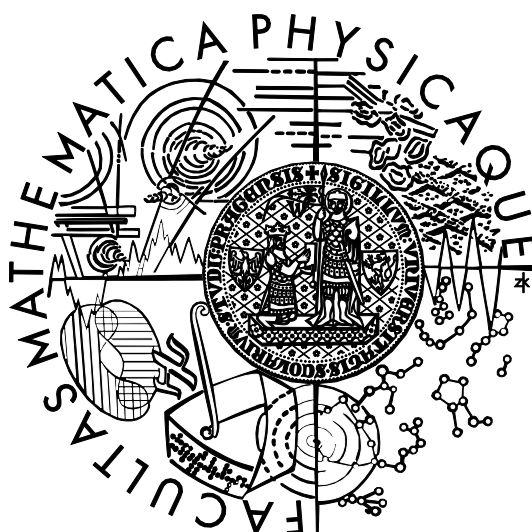


Charles University in Prague
Faculty of mathematics and physics

MASTER THESIS



Viliam Šimko
Multimedia Grid

Department of software engineering
Leader of the thesis: RNDr. Leo Galamboš, Ph.D.
Study program: Informatics, Software systems

The acknowledgements belongs to the CIANT organisation, especially to Mr. Pavel Smetana, Mr. Pavel Sedlák, Ms. Denisa Kera, Mr. Jan Šebek due to their support during the creation of AccessGrid node, to Mr. Don Foresta, Mr. Petr Holub and Mr. Greszieck Szedeck due to their willingness to share knowledge in the field of videoconferencing technologies, to Mr. Leo Galamboš due to the introduction of the great \LaTeX system, which has dramatically simplified the creation of this document, to Mr. Derek Piper for his superb AGVCR tool and to L'ubka for her mental support, patience and all the goodies she used to bring me in front of my computer.

Hereby I declare that I wrote this thesis independently using exclusively the cited sources listed in bibliography. **Súhlasím s vypožičiavaním práce.**

Viliam Šimko, October 26, 2007, Prague

Contents

1	Introduction	1
2	Výber technológií	3
2.1	Motivácia	3
2.2	Vybrané komerčné systémy	4
2.2.1	Skype	4
2.2.2	Adobe Flash	5
2.2.3	InSORS	6
2.2.4	VRVS + EVO	7
2.3	Vybrané voľne dostupné systémy	9
2.3.1	VoIP riešenia	9
2.3.2	AccessGrid	10
2.4	Požiadavky a parametre zvoleného riešenia	12
2.4.1	Základné požiadavky	12
2.4.2	Obraz	14
2.4.3	Zvuk	15
2.4.4	Prezentácia/Vizualizácia	15
2.4.5	Spolupráca	16
2.5	Vybraný systém	17
3	Protocols	18
3.1	Multicast	18
3.1.1	IGMP	19
3.1.2	SAP	20
3.1.3	MADCAP	21
3.1.4	UMTP	21
3.2	Multimedia-related protocols	22
3.2.1	RTP and RTCP	22
3.2.2	RSVP	23
3.2.3	RTSP	23
3.2.4	SDP	23
3.3	VoIP protocols	24
3.3.1	SIP	24
3.3.2	H.323	24
3.3.3	IAX2	24
3.4	Evaluation of selection of protocols	25

4	Codecs	26
4.1	Need of codecs	26
4.2	Video and audio transmission using RTP protocol	27
4.2.1	Video: NV	27
4.2.2	Video: H.261	27
4.2.3	Video: H.263	27
4.2.4	Video: H.264 (MPEG-4 AVC)	28
4.2.5	Video: MPEG-2	28
4.2.6	Video: Motion JPEG	29
4.2.7	Audio: L16	29
4.2.8	Audio: GSM	29
4.2.9	Audio: G.722	29
4.2.10	Audio: Speex	29
4.3	Evaluation of selection of codecs	30
5	CIANT AccessGrid node	31
5.1	Series of Polylogue video-conferences	31
5.2	AccessGrid in the art community	32
5.3	Information portal	33
5.4	Components used during the node setup	34
5.4.1	AG-Display	34
5.4.2	AG-AudioVideo	35
5.4.3	AG-Bridge	36
5.4.4	Gentner AP-800	36
5.5	Videoconferencing room	36
5.6	Day-to-day operation experience	38
6	NAT/Firewall	40
6.1	NAT-related issues	41
6.2	A simple NAT-traversal test	41
6.3	Tunneling	42
6.3.1	NAT-to-NAT communication	42
7	Linux-only AccessGrid	44
7.1	Emplacement	44
7.2	Network connectivity	44
7.3	Hardware	45
7.4	Operating system and essential software	45
7.5	Video and audio transmission	46
7.5.1	OpenMash framework	46
7.5.2	VIC	47
7.5.3	RAT	48
7.6	Shared applications	48
7.6.1	Shared presentations	49
7.6.2	Shared desktop	50
7.6.3	Shared media player	50
7.6.4	Further shared applications	51

7.7	G-Ware configuration tool	51
7.8	Bridging the multicast	51
7.8.1	QuickBridge	52
7.8.2	AGConnector	54
7.9	Results and suggestions	54
8	Recording the conference	55
8.1	Recording vs archivation	55
8.2	Data-streams within the AccessGrid session	55
8.2.1	Candidates for archivation	56
8.2.2	Conversion to common format	56
8.3	Existujúce nástroje pre nahrávanie	57
8.3.1	Tcpdump – Packet sniffer	57
8.3.2	RTPtools	57
8.3.3	AGVCR – RTP recorder/player	57
8.4	Nahrávanie prebiehajúcej konferencie	58
8.5	StreamVNC – Nový nástroj pre lokálne nahrávanie	58
8.5.1	Otázka nahrávania RTP paketov	59
8.5.2	Otázka spracovania zvuku	59
8.5.3	Otázka spracovania obrazu pracovnej plochy	59
8.5.4	Otázka spracovania obrazu z kamery	60
8.5.5	Podpora prezentačných nástrojov	60
8.5.6	Architektúra StreamVNC	60
8.5.7	Implementované profily	61
8.5.8	Správa bežiacich procesov	62
8.5.9	Test konzistencie prostredia	63
9	Záver	64

Abstract

Title: Multimedia Grid

Author: Viliam Šimko

Department: Department of Software Engineering

Supervisor: RNDr. Leo Galamboš, Ph.D.

Supervisor's e-mail address: `leo.galambos@mff.cuni.cz`

Abstract: The thesis describes the possibilities and issues of existing videoconferencing systems. We focus on open-source solutions for group-to-group communication especially the AccessGrid technology that utilizes the multicast packet delivery method of multimedia content. We describe AccessGrid node created for CIANT organization, which has been taken as a starting point for defining the architecture of a videoconferencing system based exclusively on open-source components, particularly Linux and related software. We address issues related to firewalls and NAT traversal and provide a comparison of various videoconference-related codecs and protocols. Finally, the text touches the question of recording and archiving of conference sessions and provides a new integrated recording application – StreamVNC. The tool allows users to prepare their presentations using localhost recording facility which includes AGVCR for recoding RTP packets, EVIC for grabbing the VNC display and camera source and RAT as an audio encoder.

Keywords: AccessGrid, CIANT, RTP, MBone, VIC, RAT, VRVS, VNC, Videoconference, SharedPresentation, StreamVNC, multicast, sharing, communication, open-source, recording, archiving

Chapter 1

Introduction

The dawn of videoconferencing technologies is tightly coupled with the invention of television broadcast. Even NASA utilized two independent radio frequencies as audio-visual signal carrier in both directions during their first space missions. The market brought the first video-phones and video-conferencing services later on. These services, however, mostly required purchase of inflexible devices. The expansion of cheap videoconferencing systems was only possible when the fast digital networks in the late eighties were introduced (for example 128Kbit/s ISDN services). Later, in the early nineties, the progressive development of the Internet and its IP protocol laid the foundations of flexible videoconferencing technologies. **The world of telecommunications lost their exclusive position in the field of multimedia transmissions.** More historical-related information is available in the literature [51], [44].

TRANSLATE: V dobe vzniku tohto textu sa bežne stretávame s takzvanou *desktop-to-desktop* komunikáciou: text messages exchange using common chat applications such as IRC, ICQ or Jabber, transmission of sound and image using applications such as Skype or MS NetMeeting, which even provide conferencing functionality among multiple users. There are, however, cases when these services simply do not suffice, when groups become participants of a communication – so called *group-to-group* communication. The differences appear immediately in increased hardware and software demands. Each group usually makes use of multiple cameras

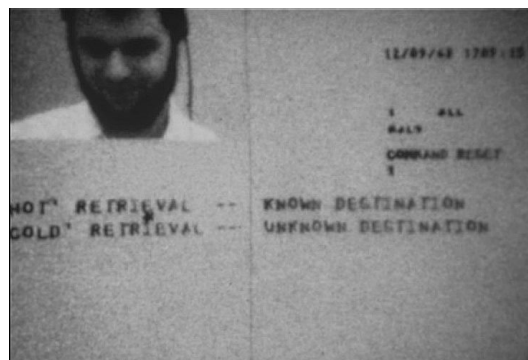


Figure 1.1: Videoconferencing system 1968. (source [51])

and other visualization tools. Sound quality within the conferencing rooms plays an important role in this context. The participants have to deal with sound feedback and other intrusive effects.

TRANSLATE: Nech už sa jedná o skupiny vedeckých pracovníkov, členov rôznych teamov ale aj o komunikáciu medzi vyučujúcimi a študentmi, the most important issue always touches the simplicity, availability and affordability of the particular technology. Nowadays, remote scientific workplaces as well as universities dispose of high-speed computer networks. TRANSLATE: Ponúka sa teda možnosť využiť existing infrastructure for videoconferencing purposes as well. As an example we could point out the Masaryk University in Brno, where this approach has been adopted in the implementation of videoconferencing node called AccessGrid Point. The idea has been further described in paper [24], where the authors E. Hladká and P. Holub refer to the possibility of extension of computational grids¹ using devices enabling virtual meetings (videoconferences). This principle has also been popular among founders of grid-related projects thus along with the grid infrastructure also the AccessGrid network, composed of special videoconferencing rooms, has been developed. This rooms have been equipped to facilitate distant high-quality collaboration.

Pre nás je podstatné, že okrem množstva komerčných aplikácií existujú alternatívne riešenia založené na open-source nástrojoch. Hlavným problémom však ostáva jednoduchosť inštalácie a používania. Videokonferenčné systémy majú obvykle vyššie nároky na užívateľov, správcov aj použitý hardvér. Veľakrát je pre užívateľa jednoduchšie zakúpiť drahé zariadenie, ako prispôsobiť open-source aplikáciu svojim požiadavkom².

V práci boli stanovené dva základné ciele:

1. Nájsť spôsob, ako s využitím výhradne open-source technológií vytvoriť funkčný videokonferenčný uzol pre group-to-group komunikáciu. Systém bude zameraný na Unixové prostredie, konkrétne platformu Linux.
2. Popísať možnosti nahrávania, úpravy a zálohovania konferencie.

Ústredným bodom nášho záujmu bude technológia AccessGrid. Pokúsime sa ju porovnať s inými dostupnými technológiami, poukázať na jej výhody a nevýhody a ponúknuť návod na jej budúce vylepšenie. Pomocou existujúcej videokonferenčnej miestnosti priblížime túto technológiu bežným užívateľom.

¹Grid – distributed system which provides facilities to solve computationally complex problems using parallel algorithms such as the problems concerning physics of high energies, astrophysics or biology.

²Úvahu o problémoch open-source softvéru ponecháme na čitateľovi.

Chapter 2

Výber technológie

2.1 Motivácia

Zadanie tejto práce zámerne vylučuje využitie iného ako open-source software. Chceme navrhnúť voľne dostupný modifikovateľný systém a preto potrebujeme prístup k zdrojovým kódom všetkých použitých komponentov. Problémom komerčných produktov však nie je iba cena, ale aj v obtiažnosť portovania na nové platformy, v našom prípade na platformu Linux. (Samozrejme neuvažujeme rôzne formy emulácie typu VMWare, či Wine). Ďalšou nevýhodou uzavretých systémov sú uzavreté protokoly, čo sťažuje pridávanie a úpravu vlastností do existujúceho systému.

Našou snahou je teda využívať vhodné štandardy akými sú napríklad IETF¹ protokoly. IETF je organizácia, ktorá sa zaoberá vývojom štandardov používaných v rámci internetu. Členstvo v tejto organizácii je dobrovoľné a celý standardizačný proces prebieha verejne. Množstvo informácií môžeme čerpať z tzv. RFC² dokumentov, ktoré vznikli za prispenia komunity a odborníkov zo špecifických oblastí podľa zamerania daného dokumentu. Z RFC budeme čerpať napr. informácie o spôsobe prenosu obrazu a zvuku nad protokolom IP.

Napriek všetkým nevýhodám uzavretých technológií sa necháme inšpirovať ich vlastnosťami. Poskytnú nám cenné informácie o použiteľnosti a ergonómii. Produkty akými sú VRVS [13], či InSORS sú dokonca priamo vystavané na základoch voľne dostupných komponentov (napr. Mbone tools). Môžeme tak priamo využiť niektoré ich časti vo vlastnom projekte.

Nakoniec, ako uvádza [50], si musíme uvedomiť, že požiadavky na videokonferenčný systém, ktoré má vedecká komunita sa budú líšiť od požiadaviek komerčného sektora.

V krátkosti popíšeme niekoľko vybraných aplikácií, pričom spomenieme vždy ich najdôležitejšie výhody a nevýhody. Nasledujúci výber je založený na osobných skúsenostiach a preferenciách autora, samozrejme s ohľadom na praktickú použiteľnosť.

¹Internet Engineering Task Force

²Request For Comments

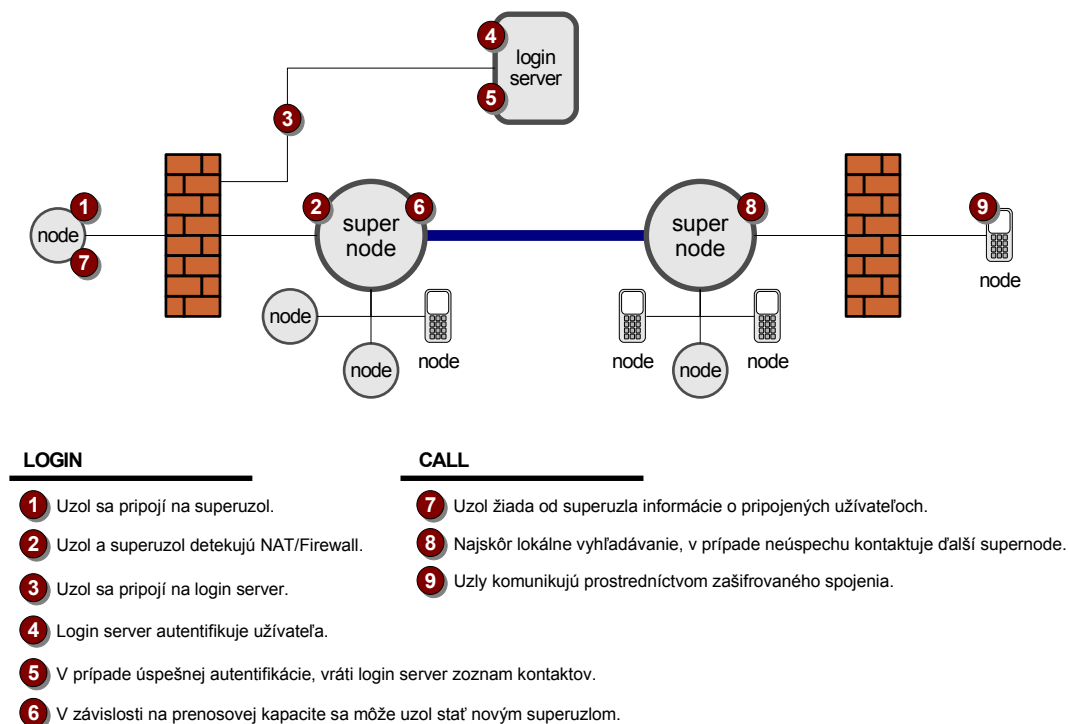


Figure 2.1: Architektúra systému Skype.

2.2 Vybrané komerčné systémy

2.2.1 Skype

Systém Skype sme vybrali ako zástupcu komerčne úspešného a masovo používaného systému s jednoduchou inštaláciou a obsluhou.

Obrázok 2.1 znázorňuje spôsob komunikácie medzi uzlami v sieti Skype. Uzly umiestnené za sieťovým firewallom komunikujú s okolitým svetom prostredníctvom tzv. superuzlov³. Prihlasovanie do siete sa uskutočňuje pomocou login servera. Po úspešnej autentifikácii sprostredkujú superuzly multimediálny prenos medzi koncovými klientmi. Nadväzovanie spojenia prebieha tak, že klient postupne kontaktuje najbližšie superuzly, ktoré rekurzívne kontaktujú svojich susedov až pokiaľ nie je známa trasa k volanému klientovi. Výsledné smerovanie multimediálnych paketov je zabezpečené na aplikačnej úrovni (podľa modelu ISO/OSI) a využíva sa decentralizovaná povaha siete Skype.

Spomínaná závislosť na superuzloch je priamo spojená s uzavretosťou protokolu Skype. Ako vidíme na obrázku 2.1, klient, ktorý disponuje dostatočne rýchlym pripojením a vlastnou IP adresou, sa mení na tzv. superuzol. Ako uvádza zdroj [35], pokiaľ by existovali alternatívne rozhrania, mohli by takéto aplikácie zabráňovať vzniku superuzlov, čo by sa následne odrazilo na celkovej kvalite prenosu v celej sieti Skype. Autori preto nemajú záujem na zverejňovaní svojho protokolu.

Automatický vznik superuzlov vedie k využívaniu prenosovej kapacity bez vedomia vlastníka siete. Problém sa prejaví pri spoplatnenom objeme prenesených dát,

³V ďalšom texte uvidíme ešte iné prístupy k riešeniu problémov spojených s firewallmi

v Českej Republike napr. v sieti rozvodov káblovej televízie a internetu UPC alebo s pripojením ADSL.

Výhody

- Jednoduchá inštalácia a obsluha.
- Distribuovaná povaha aplikácie umožňuje prekonať NAT/firewall pomocou siete superuzlov (obrázok 2.1).
- Aplikácia je v súčasnosti zdarma, poskytovateľ umožňuje aj telefonické hovory do komerčných sietí.
- Aplikácia je k dispozícii pre platformy Windows, Mac, Linux a Pocket PC.
- Možnosť uskutočniť videokonferencie pomocou ďalších modulov – text, audio, video, selektívne zdieľanie pracovnej plochy.
- Komunikácia medzi uzlami je šifrovaná. Výmena kľúčov je zabezpečená pomocou 2048-bit RSA šifry a samotný prenos využíva 256-bit AES.

Neýhody

- Uzavretý protokol.
- Rozhranie pre platformu Linux zatiaľ vo verzii 1.3.0.53 neumožňuje prenos obrazu.
- Vznik superuzlov, využívanie prenosovej kapacity bez vedomia vlastníka siete.
- Aplikácia neumožňuje ovplyvňovať kvalitu prenosu priamo, nastavenie prebieha automaticky.
- Systém nie je možné používať bez internetového pripojenia (napr. v izolovanej sieti LAN). Vždy je nutné kontaktovať Login server (obrázok 2.1).

2.2.2 Adobe Flash

Technológia Flash vzhľadom na svoje rozšírenie vo väčšine súčasných internetových prehliadačov môže ponúknuť spôsob ako dopraviť videokonferenčné riešenie k užívateľovi bez nutnosti inštalácie. Najnovšia verzia Flash 9 je dostupná pre platformy Windows, Mac a Linux, staršie verzie existujú pre platformy Solaris, HP-UX, BeOS a ďalšie. Ako príklad videokonferenčných riešení postavených na tejto technológii môžeme uviesť InstantPresenter [17] alebo FlashMeeting [16].

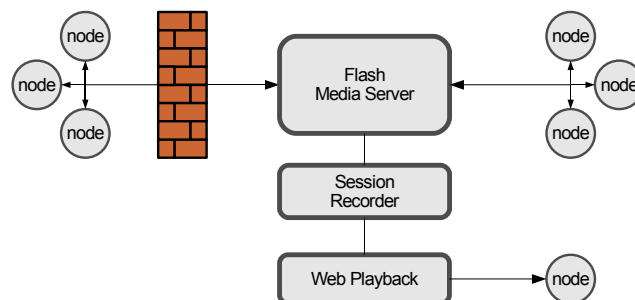


Figure 2.2: Architektúra Flash Media Server.

Na obrázku 2.2 je znázornená centralizovaná povaha systému. Predpokladá sa, že centrálny prvok Flash Media Server disponuje verejnou IP adresou, a dostatočne rýchlym pripojením aby zvládla spracovať komunikáciu pripojených klientov (na obrázku označených ako „node“). Takto centralizovaná architektúra umožňuje prekonávať NAT/firewall, pretože dokáže v krajnom prípade zabaliť komunikáciu do TCP paketov.

Prenos multimediálnych paketov zabezpečuje proprietárny Real Time Messaging Protocol (RTMP⁴). Pretože neexistuje voľne dostupná špecifikácia RTMP, vznikol open-source projekt Red5 [18], ktorý pomocou reverse engineeringu⁵ implementuje jeho podstatnú časť [2]. Protokol RTMP využíva UDP port 1935 a v prípade blokovania tunel cez TCP port 80 (HTTP).

Výhody

- Technológia dostupná pre rôzne platformy.
- Pokiaľ odhliadneme od nutnosti inštalovať samotný internetový prehliadač a flash-plugin, nevyžadujú ďalšie aplikácie žiadnu inštaláciu. Komunikácia prebieha priamo z internetového prehliadača.
- Jednoduché nastavenie a kalibrácia kamery a mikrofónu.
- Ďalšie vlastnosti, ktoré uvádza výrobca, ako napr. komunikácia cez SSL, vyvažovanie záťaže, QoS.
- Množstvo open-source projektov súvisiacich s technológiou flash [15].

Neýhody

- Systém je určený pre relatívne malé konferencie vzhľadom na vyššie hardwarové nároky samotného flash-pluginu v porovnaní napr. s VRVS alebo AccessGridom popisovaných neskôr v tomto texte.
- Zatiaľ neexistuje plnohodnotná open-source implementácia ani iná natívna implementácia pre architektúry x86-64, PowerPC, ARM.
- Závislosť na reverse engineeringu.

2.2.3 InSORS

Riešenie InSORS IG2 je postavené na technológii Mbone o ktorej budeme hovoriť v ďalších kapitolách v súvislosti s AccessGridom (kapitola 2.3.2). InSORS prináša integrované prostredie, ktoré zahŕňa softwarové aj hardwarové komponenty. Zákazník tak môže získať vyladený videokonferenčný systém, ktorý je ale zatiaľ dostupný iba pre platformu Windows.

Podobne ako v prípade AccessGridu, prebieha komunikácia buď prostredníctvom multicastovej siete, alebo pomocou premostení z unicastovej siete (obrázok 2.3).

Systém disponuje nástrojom IG Recorder, ktorý umožňuje okrem zvuku a obrazu nahrávať zdieľané prezentácie. Záznamy z konferencií je možné zároveň prehrať v prebiehajúcej videokonferencii.

⁴Nezamieňať s Routing Table Maintenance Protocolom z rodiny AppleTalk.

⁵Autori označujú tento akt ako „discovery through observation“.

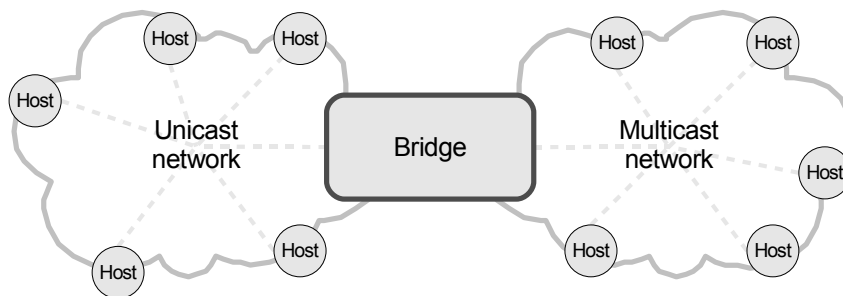


Figure 2.3: Architektúra systému s použitím multicastovej siete.

Výhody

- Nahrávanie zdieľaných konferencií a prezentácií.
- Integrované prostredie – možnosť zakúpiť zároveň software aj hardware.
- Integrácia s ostatnými VoIP technológiami, premostenie systémov H.323 a AccessGrid.
- InSORS využíva otvorené štandardy ako napr. kodeky H.261, H.263, H.264 (kapitoly 4.2.2, 4.2.3, 4.2.4). Prenos multimediálnych dát je riešený pomocou protokolu RTP (kapitola 3.2.1).

Neýhody

- Aplikácia je k dispozícii jedine pre platformu Windows.
- Videokonferencia pomocou InSORS je spoplatnená.

2.2.4 VRVS + EVO

Posledným zástupcom komerčných videokonferenčných riešení, ktorý v tejto práci popíšeme, je Virtual Rooms Videoconferencing Systems (VRVS) a jeho následník EVO. VRVS je distribuovaný systém určený pre komunikáciu v reálnom čase prístupný cez webové užívateľské rozhranie, ktoré využíva platformovo závislé aplikácie na strane klienta. Ide o upravené verzie MBone nástrojov akými sú napr. VIC (kapitola 7.5.2) a RAT (kapitola 7.5.3) z projektu OpenMash (kapitola 7.5.1). Ďalšie komponenty, ako napr. premostenia pre softwarové alebo hardwarové H.323 klienty, používajú technológiu Java.

VRVS disponuje sieťou reflektorov – serverov rozmiestnených na rôznych miestach sveta, ktoré sprostredkovávajú komunikáciu medzi VRVS klientmi. Špeciálnym prípadom takéhoto reflektora je VRVS-AG Reflector, ktorý umožňuje komunikovať s uzlami AccessGridu. Nezáleží, či klient disponuje unicast alebo multicast pripojením, VRVS transparentne premost'uje technológie AccessGrid, QuickTime a H.323.

Monitorovanie stavu VRVS siete umožňuje upravená verzia systému MonaLisa⁶. Jeho grafické rozhranie vidíme na obrázku 2.4. Vľavo je na svetovej mape znázornená sieť reflektorov umiestnených v Európe, vpravo je tá istá sieť znázornená vo forme prehľadného grafu.

⁶Monitoring Agents using a Large Integrated Services Architecture

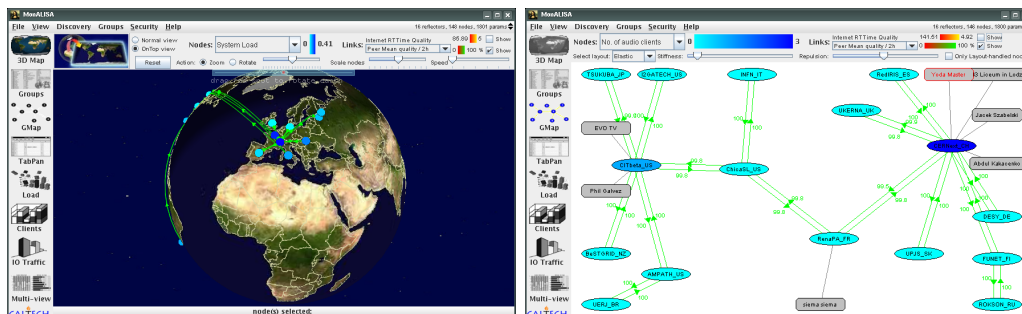


Figure 2.4: Monitorovanie distribuovaného systému VRVS reflektorov pomocou aplikácie MonaLisa.

Zaujímavou funkciou, ktorá minimalizuje nároky na prenosovú kapacitu, je funkcia „Voice Switched mode“. Autori VRVSV uvádzajú v publikácii [13] nasledujúci príklad:

Príklad: Garantovaná kapacita siete je 1024Kbit/s. Videokonferencie sa zúčastňuje 15 partnerov. Každý vysiela kvalitný obraz (s datovými tokmi okolo 800Kbit/s). Kapacita siete by teda mala byť aspoň 12Mbit/s:

Počet účastníkov x 800 Kbit/s + 64Kbit/s (minimálny audio tok) = =
(15 x 800 + 64) Kbit/s = **12064Kbit/s**.

Riešenie: Použijeme „Voice Switched“ mode, takže vytvoríme požiadavku na VRVS reflektor, aby posielal len video tok od toho partnera, ktorý práve rozpráva. Splnením tejto požiadavky dosiahneme obmedzenie toku na úroveň:

$$1 \times 800\text{Kbit/s (video)} + 64\text{Kbit/s (audio)} = \mathbf{864\text{Kbit/s.}}$$

Podobne ako InSORS a AccessGrid, podporuje VRVS aj nahrávanie konferencií. To však prebieha na strane servera, kde poverený operátor manuálne ovláda nahrávanie RTP paketov z vybranej virtuálnej miestnosti.

Aj keď ide o komerčný projekt California Institute of Technology, je systém dostupný zdarma pre vedeckú komunitu. Rozširovanie siete VRVS reflektorov je spravované autormi systému. Po splnení kritérií môže záujemca poskytovať svoj vlastný reflektor pripojený do VRVS siete. Užívatelia sú združení do skupín podľa oblasti záujmu, tzv. VRVS komunity (Internet2, SLOVAKIA...)

V septembri 2003, bola otvorená Slovenská akademická videokonferenčná sieť, ktorá je k dispozícii všetkým univerzitám, výskumným ústavom SAV a vládnym organizáciám pripojených na optickú sieť SANET (Association of users of Slovak Academic data NETwork). Vznikol tak priestor pre experimentálne projekty ako napr. otvorené alebo dištančné vzdelávanie.

Nasledovníkom VRVS sa má v blízkej budúcnosti stať aplikácia EVO. Rozhranie je vytvorené v Jave a oproti VRVS podporuje navyše: Instant Messaging, Private

Meetings (videokonferencia bez virtuálnej miestnosti), kryptovanie textových správ, výmenu súborov atď. Aplikácia je tiež multiplatformná.

Výhody

- Dostupnosť priamo z webového rozhrania – automatická inštalácia.
- Integrovaný systém rezervácie virtuálnych miestností.
- P2P-sieť reflektorov.
- Premostenie rozdielnych technológií.
- Monitorovanie stavu siete.
- Voice Switched mode.

Neýhody

- Nutná výstavba siete reflektorov.
- Užívateľ musí inštalovať Javu, potvrdzovať certifikáty.

2.3 Vybrané voľne dostupné systémy

2.3.1 VoIP riešenia

Z tejto kategórie vyberieme nasledujúce aplikácie: **Linphone**, **Ekiga**, **Wengophone** a **Asterisk**. Všetky existujú pre platformy Windows aj Linux, sú dostupné pod licenciou GPL⁷, podporujú protokol SIP, prípadne H.323 (kapitola 3.3).

Aplikácia Linphone umožňuje iba prenos zvuku. Ekiga a Wengophone využívajú pripojenú webkameru na prenos obrazu. Navyše poskytujú pre nových užívateľov bezplatný SIP účet `username@ekiga.net` resp. `username@voip.wengo.fr`. Wengophone dokonca poskytuje kredit pre volanie do pevnej a mobilnej telefónnej siete, prípadne prepája hovory priamo na SIP účet. Podporuje aj posielanie SMS správ.

Prenos obrazu je v oboch aplikáciách vo veľkosti CIF (obrázok 2.7), každý uzol môže odosielať iba jeden obrazový a jeden zvukový dátový tok, čo znevýhodňuje nasadenie týchto aplikácií pre účely rozsiahlej videokonferencie, naopak postačuje pre účely desktop-to-desktop komunikácie.

Okrem SIP a H.323 klientov by sme radi spomenuli software určený pre chod telefónnej ústredne (PBX). Asterisk podporuje prenos zvuku pomocou niekoľkých protokolov (IAX2, SIP, H.323), voicemail, adresárové služby, konferenčné hovory, odkazovač, prijímanie spojení čakajúcich vo fronte a ďalšie Design protokolu IAX2 umožňuje prekonávať problémy s NAT/firewallmi. Protokol SIP tiež podporuje preposielanie multimediálnych paketov cez tzv. STUN proxy server (napr. `stun.ekiga.net`). Nie je však natoľko úspešný ako práve IAX2.

Výhody

- Existencia pre rôzne platformy, open-source, SIP je IETF štandard. IAX2 je vo fáze návrhu (IETF Internet-Draft [56]).
- Jednoduchá inštalácia a používanie.

⁷General Public License \Rightarrow open-source

- Design IAX2 a STUN server pre SIP riešia problémy s NAT/firewallom.
- Integrácia s telefónnymi linkami.
- Kvalitné a efektívne kodeky ako napr. Speex.
- Vstavané odstraňovanie spätnej väzby.
- Pre Asterisk existuje špeciálna distribúcia Linuxu, dokonca aj vo formáte VMWare appliance.

Neýhody

- Riešenia primárne určené pre telefonovanie.
- Fixná veľkosť obrazu CIF a fixný počet dátových tokov obmedzuje použitie na desktop-to-desktop komunikáciu. Je to však problém konkrétnej implementácie, nie architektúry.
- Kombinácia rôznych SIP klientov častokrát vedie k chybám pri spájovaní a komunikácii – voľba nesprávneho kodeku, rozdielne implementácie protokolov.

2.3.2 AccessGrid

Cieľom projektu AccessGrid (AG) je umožniť čo najprirodzenejšiu interakciu medzi účastníkmi konferencie s využitím existujúcej sieťovej infraštruktúry. Tento robustný nástroj pozostáva zo sady softwarových a hardwarových komponentov škálovateľných na mieru potrebám videokonferencie, od komunikácie medzi jednotlivcami až po celé skupiny.

Po softwarovej stránke tu môžeme nájsť komponenty určené na prenos obrazu, zvuku, zdieľanie prezentácií, vizualizačné a riadiace programy. Pre konferenciu prostredníctvom AG je obvykle vyhradená špeciálna miestnosť, čo umožňuje zabezpečiť kvalitnú komunikáciu aj pri veľkom počte účastníkov. Technológia AG je priamo navrhnutá pre spoluprácu medzi skupinami vedcov, kde prezentácia a vizualizácia výsledkov výskumu vyžaduje viac projekčnej techniky, kamier, náročnejšie spracovanie zvuku, a teda aj väčšie množstvo výpočtovej techniky potrebnej pre kódovanie zvuku a obrazu.

Obvykle takúto miestnosť označujeme ako uzol AccessGridu (AG node). Obrázok 2.5 znázorňuje príklad typickej architektúry AG node. Hneď na prvý pohľad si môžeme všimnúť distribuovanú povahu celého systému. Na obrázku sú znázorené štyri oddelené počítače – Display Computer, Video Capture Computer, Audio Capture Computer a Control Computer. Po softwarovej stránke je AG zložený z komponentov VenueServer, VenueClient, NodeManager, NodeService a Bridge. Venue server predstavuje virtuálnu miestnosť, v ktorej sa nachádzajú účastníci konferencie. Obsahuje súbory určené pre stiahnutie zdieľanými aplikáciami. Na VenueServer sa klienti pripájajú pomocou svojich VenueClientov, kde sa každý účastník autentifikuje pomocou certifikátu⁸. NodeManager umožňuje ovládať distribuované prvky systému, zapínať a vypínať streamovacie služby. To sa deje prostredníctvom NodeService – služieb spájajúcich oddelený hardware do distribuovaného systému,

⁸O certifikát musí účastník dopredu požiadať certifikačnú autoritu. K tomuto účelu slúži špeciálny nástroj pomenovaný „Certificate Request Tool“. V prípade nutnosti je možné použiť anonymný certifikát.

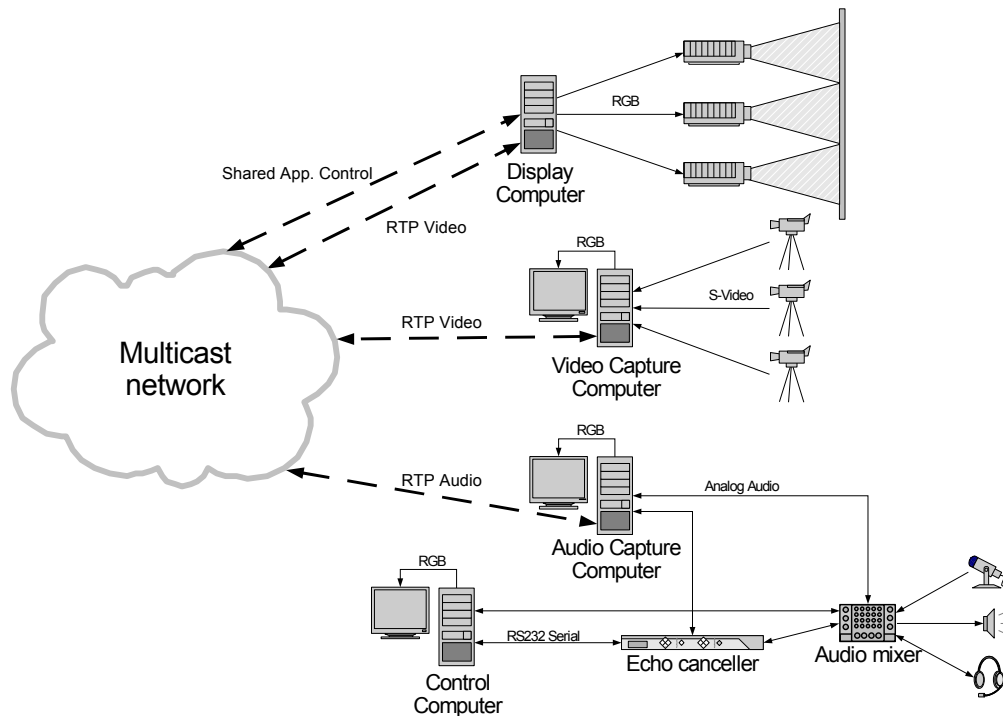


Figure 2.5: Typická architektúra AccessGridu. (zdroje [24][8])

ktorý komunikuje pomocou XML-RPC. Posledným komponentom je Bridge, ktorý je určený na sprostredkovanie služby pre účastníkov bez multicastovej siete.

V porovnaní s ostatnými technológiami sa môže zdať AG ako príliš zložitý systém. Kvalitný zážitok z videokonferencie za účasti veľkého počtu osôb, však vyžaduje viac investícií do zariadenia miestnosti. Obraz je na rozdiel od monitoru zobrazovaný na projekčné plátno niekoľkými projektormi.

Výhody

- Škálovateľné riešenie podľa veľkosti konferencie.
- Množstvo zdieľaných aplikácií a iných nástrojov.
- Vysoká kvalita obrazu a zvuku.
- Možnosť prenosu prostredníctvom unicast aj multicast siete.
- Multiplatformnosť: Linux, Windows, MacOS X, FreeBSD

Neýhody

- Architektúra vyžaduje rýchle sieťové pripojenie.
- Bežný užívateľ môže mať problémy pri inštalácii a nastavení AG.
- Vyššie nároky na priestor a vyššie finančné náklady na nákup potrebného hardware.
- Problémy s multicastovým pripojením.
- Problémy s prekonávaním NAT.

2.4 Požiadavky a parametre zvoleného riešenia

Predchádzajúci výber aplikácií nebol náhodný. Z komerčných riešení sme vybrali Skype pre jeho masové rozšírenie v oblasti IP telefonie a jeho schopnosť prekonávať NAT/firewall pomocou distribuovaného systému superuzlov. Adobe Flash sme vybrali pre jeho dostupnosť vo väčšine internetových prehliadačov a jednoduchosť inštalácie. InSORS nás oslovil spôsobom, ktorým využíva technológiu MBone a ďalšie otvorené štandardy, ktoré ponúka v ľahko ovládateľnom balíku s možnosťou zakúpenia spolu so špecializovaným hardwarom. Systém VRVS je pre nás príkladom aplikácie spustiteľnej priamo z webu (teda bez nutnosti inštalácie) a príkladom distribuovanej siete reflektorov monitorovaných systémom MonaLisa (obrázok 2.4). Z nekomerčných aplikácií sme vybrali Wengophone a Ekigu ako zástupcov SIP a H.323 a AccessGrid ako zástupcu robustného open-source videokonferenčného riešenia.

Cieľom nasledujúceho textu je definovať zoznam požiadaviek, ktoré kladieme na výsledný systém na základe skúseností z existujúcich videokonferenčných riešení. V skutočnosti nemusí existovať systém, ktorý by vyhovoval všetkým požiadavkom, stačí ak sa tomuto ideálu priblíži.

2.4.1 Základné požiadavky

Škálovateľnosť: Architektúra videokonferenčného systému by mala byť dostatočne flexibilná na to, aby umožňovala zároveň komunikáciu medzi veľkým aj malým počtom účastníkov, tým sa otvorí možnosť nasadenia v organizáciách nezávisle na ich veľkosti. Veľké organizácie budú mať možnosť nasadiť distribuovanú verziu systému, zatiaľ čo malé organizácie vystačia so základným nastavením. Príliš zaťažný systém by mal byť jednoducho rozšíriteľný o nové hardwarové komponenty. Túto myšlienku by sme mohli rozvinúť až tak ďaleko, že budeme uvažovať o vytvorení virtuálnej univerzity s množstvom simultánne prebiehajúcich prednášok.

Bezpečnosť: Umožniť šifrovanú komunikáciu v rámci konferencie. V ideálnom prípade úplné šifrovanie všetkých správ. Možnosť voľby miery zabezpečenia. Ďalšou funkčnosťou môže byť autentifikácia a autorizácia účastníkov tak, aby mali prístup do vybranej podmnožiny prebiehajúcich konferencií. V prípade bezpečnosti budeme taktiež požadovať, aby dátové prenosy nemohli byť modifikované neznámymi subjektmi (integrita). Pre niektoré organizácie býva práve táto požiadavka kľúčová.

Dostupnosť: Hlavné vyriešenie problémov spojených s firewallmi, NAT a proxy servermi. Množstvo klientov sa nachádza za istou formou NAT kvôli problémom adresovania v IPv4 sieti. Sieť býva navyše obohatená o firewall, prípadne rôzne formy tzv. transparent proxy. Dokonalý prechod cez všetky sieťové prekážky by bol veľmi komplikovaný, ale systém by mal počítat aspoň s najbežnejšími problémami, hlavne NAT.

Modifikovateľnosť: Možnosť zmeny komponentov systému.

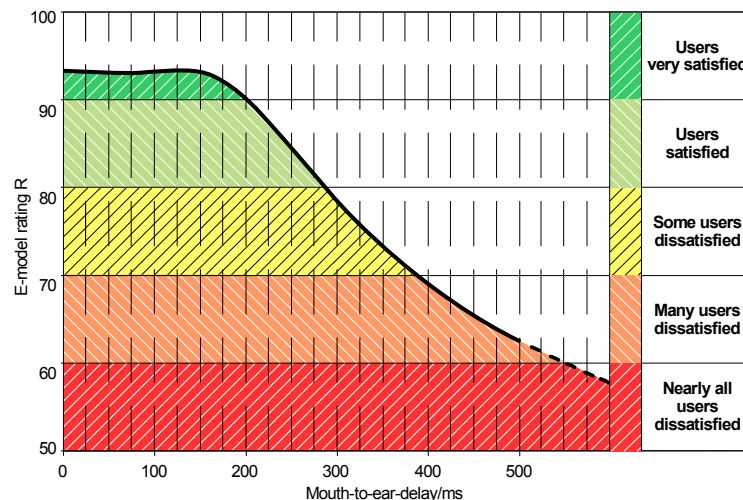


Figure 2.6: Dôsledky oneskorenia zvuku podľa ITU-T G.114 [29].

Stabilita: Stabilita v čase. Systém by mal byť schopný bežať nepretržite bez pádu. V prípade poruchy by mal zanechať konzistentnú množinu dát.

Prenosová kapacita: Úspech videokonferencie výrazne závisí na rýchlosti siete, ktorou účastníci disponujú.

One-way Latency: Jednosmerná reakčná odozva je doba medzi odoslaním paketu na strane odosielateľa a prijatím paketu na strane príjemcu. V ďalšom texte nás bude hlavne zaujímať hodnota Round-trip Latency.

Round-trip Latency: Túto veličinu definujeme ako dobu medzi odoslaním požiadavku a prijatím odpovede ($\approx 2 \times \text{onewaylatency}$). V texte ju budeme používať pod názvom **Reakčná odozva** resp. **Latency**. V literatúre [29] sa uvádza, že v prípade zvuku by nemala presahovať 300ms, inak si účastníci rozhovoru prestávajú rozumieť. Znázorňuje to aj obrázok 2.6. V prípade prenosu obrazu môže byť táto hodnota bez problémov niekoľkonásobne väčšia. V multimediálnej komunikácii preto uprednostníme nespoľahlivé nepotvrdzované doručovanie tzv. **Unreliable Unicast/Multicast**. Doručovanie pomocou UDP pred TCP.

Jitter: Rozptyl reakčnej odozvy definujeme ako zmenu reakčnej odozvy v čase. Garancia maximálnej hodnoty pre jitter umožňuje odhadnúť veľkosti bufferov. Použitie bufferov je však obmedzené celkovou latenciou.

Hardwarové nároky: Zamýšľaný systém bude spracovávať desiatky nezávislých dátových tokov. Použité kodeky musia byť v tom prípade nenáročné na výpočtový výkon. Hardwarovo náročné procesy môžu zhoršiť reakčnú odozvu systému. Tento problém sa týka hlavne spracovania obrazu.

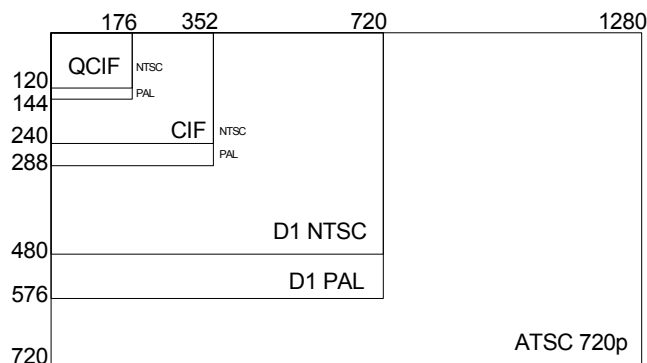


Figure 2.7: Porovnanie rôznych veľkostí obrazu.

Štandardy: Budeme používať štandardizované komunikačné protokoly a kodeky. Táto požiadavka plynie už z nášho zámeru používať jedine open-source software. Pre prenos multimediálnych paketov tak prichádza do úvahy protokol RTP (kapitola 3.2.1).

Dostupnosť pre rôzne platformy: Umožniť sledovanie konferencie minimálne na systémoch Linux a Windows.

2.4.2 Obraz

Veľkosť obrazu: Systém by mal umožniť voľbu veľkostí obrazu. Myslíme tým hlavne rozdiely medzi snímaním veľkej miestnosti s množstvom účastníkov oproti detailnému snímaniu tváre. Predpokladáme, že jednotlivé obrazové toky budeme zobrazovať na veľké projekčné plátno pomocou data-projektora, preto doporučujeme použiť aspoň formát D1 PAL 720x576 (viď obrázok 2.7). Implicitným formátom väčšiny súčasných videokonferenčných systémov býva totiž formát CIF 352x288 resp. 352x240 (viď obrázok 2.7). Toto obmedzenie sa týka hlavne použitého kodeku H.261 (pozri kap. 4.2.2).

Popisované systémy Skype a Flash (kap. 2.2.1, 2.2.2) pracujú s fixnou veľkosťou obrazu. Výhodou je jednoduchšia implementácia a ovládanie zo strany užívateľa. Naopak systémy AccessGrid, InSORS a VRVS (kap. 2.3.2, 2.2.3, 2.2.4) dávajú možnosť voľby na úkor jednoduchosti.

Kódovanie pohybu: Obraz z miestnosti, kde prebieha konferencia, môže mať statickú alebo dynamickú povahu, čo by sme mali zohľadniť aj pri voľbe vhodného kodeku. Za statický obraz v tomto prípade pokladáme napr. skupinu sediacich ľudí, ktorí sú snímaní kamerou z niekoľkometrovej vzdialenosti pri konštantných svetelných podmienkach. Príkladom dynamického obrazu môže byť napr. jedna osoba snímaná z blízka, prípadne množstvo pohyblivých objektov. Slabé osvetlenie navyše spôsobuje nárast dátového toku, pretože kamery pridávajú do obrazu rôzne artefakty.

Kvalita obrazu: Pokiaľ nám ide o komunikáciu medzi jednotlivcami, ktorí navyše disponujú prenosovou kapacitou menšou ako 1Mbit/s, prirodzene nám postačí nižšia

kvalita obrazu ako v prípade rozsiahlej konferencie, kde obvykle vyžadujeme aspoň 10 Mbit/s. V prípade požiadavku na prenosy vo vysokej kvalite musíme počítať dokonca s kapacitou 25Mbit/s pre jeden dátový tok (komprimované video vo formáte HDV). Rozpracovanie tejto problematiky je možné nájsť napr. v článku [25].

2.4.3 Zvuk

Kvalita zvuku: Na kvalitu zvuku kladieme vyššie nároky než na kvalitu obrazu. Človek je oveľa citlivejší na drobné výpadky zvuku ako na úplnú absenciu obrazu. (V kapitole 3.2.1, pojednávame o protokole RTP, ktorý práve z tohto dôvodu používa oddelené UDP porty pre prenos zvuku a obrazu). Pokiaľ chceme prenášať reč a nezaujíma nás prenos hudby v najvyššej kvalite, vystačíme s frekvenciou okolo 8000Hz. Najväčším nepriateľom pre nás bude latency (reakčná odozva) resp. jitter (rozptyl reakčnej odozvy).

Problém spätnej väzby: V podmienkach ozvučenej miestnosti sa musí systém vyrovnať so vznikom spätnej väzby a inými nevhodnými zvukovými efektmi. Potlačenie ozveny má vyššiu prioritu ako samotná kvalita zvukového signálu.

2.4.4 Prezentácia/Vizualizácia

Podporované formáty: Od poskytovateľa videokonferenčnej technológie sa očakáva, že bude v maximálnej miere vychádzať v ústrety účastníkom a podporovať široké portfólio formátov. V ideálnom prípade by účastníci nemali byť obmedzovaní pri výbere formátu, ktorý chcú prezentovať. V reálnom svete však musí poskytovateľ definovať zoznam podporovaných formátov a oboznámiť s ním všetkých zainteresovaných účastníkov s dostatočným predstihom.

Video sekvencie: Mali by sme podporovať rôzne kodeky a veľkosti od súborov s desiatkami MiB až po video uložené vo formáte DVD. Typickým požiadavkom užívateľa môže byť spustenie video sekvencie, ktorú je možné ovládať minimálne tlačítkom PLAY/PAUSE.

Automatické konverzie medzi video formátmi zvládajú napr. nástroje FFMPEG, MPlayer a VLC. Zvukové súbory môžeme konvertovať pomocou nástroja sox. Pre interaktívnu úpravu zvuku je vhodná aplikácia Audacity, v prípade obrazu poslúži napr. Avidemux.

Prezentácia vo forme statických snímok: Obvyklým formátom pre prípravu prezentácie bývajú formáty PDF a PPT. V našom riešení budeme uprednostňovať formát PDF. Účastníci videokonferencie budú totiž nútení vypustiť videosekvencie a animované prechody medzi snímkami zo svojich prezentácií. Zmenší sa tak objem prenesených dát aj záťaž procesoru.

Účastníci, ktorí inklinujú k prostrediu typu Unix budú pravdepodobne používať formáty PDF, PS (PostScript) alebo ODF (Open Document Format). Užívatelia Windows prinesú pravdepodobne formáty PPT, PDF alebo ODF. Automatickú konverziu je možné zabezpečiť jednoduchým skriptom, ktorý využíva OpenOffice makrojazyk (zdroj [12]) Ďalšie úpravy vektorových formátov pokrývajú nástroje

psstoeit, a2ping a xpdf. Automatické konverzie medzi rastrovými formátmi je možné zabezpečiť pomocou balíka NetPbm a ImageMagick. Pre vizualizáciu dát pomocou skriptov môžeme využiť programy Gnuplot, R či Graphviz.

Interaktívne 2D/3D simulácie: Do tejto kategórie patria rôzne 3D modely a prostredia vytvorené v špecializovaných aplikáciách typu 3D Studio Max, Blender, SolidEdge či AutoCad. Patria sem aj prostredia vytvorené pre niektorý z herných enginov ako napr. Unreal, Half-Life, Quake a mnoho ďalších. Medzi prostredia, ktoré by sme mohli podporovať patrí ďalej OpenInventor a VRML.

Spôsoby prezentácie by sme mohli rozdeliť do štyroch skupín:

1. Programy spustené na externom počítači, ktorý je pripojený S-Video signálom.
2. Programy používajúce VNC display.
3. Programy, ktoré priamo podporujú kolaboráciu.
4. Programy emulované pomocou VMWare.

Počas videokonferencií Polylogues sa ukázalo, že najstabilnejším riešením je použitie externého počítača pripojeného cez S-Video. Užívateľ tak môže priniest vlastné prostredie, na ktorom má odladenú svoju prezentáciu.

Obraz z externého zdroja: Už sme spomenuli, že vhodným riešením je použiť externý hardware pripojený cez S-Video do digitalizačnej karty. Nevýhodou môže byť nutnosť zakúpenia ďalšieho hardwarového komponentu. Niekedy však užívateľ nedisponuje ani S-Video výstupom, takže musíme pristúpiť napr. k inštalácii VNC.

Kvalita: Pri prenose videosekvencií spolu so zvukom platia rovnaké pravidlá ako sme stanovili v požiadavkách na obraz a zvuk. Navyše musíme brať do úvahy, že pri prezentácii snímok, ktoré majú väčšinou statickú povahu budeme používať iné techniky, kodeky, veľkosti obrazu. Situáciu mierne komplikujú prezentácie, ktoré v sebe obsahujú videosekvencie. Z toho dôvodu uprednostňujeme prezentáciu vo formáte PDF.

Ovládanie: Účastníci konferencie by mali dostať možnosť ovládať prebiehajúcu prezentáciu. V prípade AccessGridu je to napr. SharedPresentation alebo Shared-PDF.

2.4.5 Spolupráca

Riešenie by malo podporovať rôzne formy spolupráce medzi účastníkmi konferencie. Od jednoduchého zdieľania súborov až po zdieľanú pracovnú plochu. V tomto prípade môžeme znova využiť vlastnosti VNC.

2.5 Vybraný systém

V prvej kapitole sme popísali niekoľko existujúcich aplikácií, vyberali sme hlavne prakticky odskúšané riešenia z oblasti closed-source aj open-source software. Snažili sme sa popísať technológie vhodné pre desktop-to-desktop komunikáciu ako aj pre group-to-group komunikáciu a poukázali sme na rozdiely medzi oboma prístupmi. Pre každú aplikáciu sme pripravili zoznam ich výhod a nevýhod. Vytvorili sme obraz ideálneho videokonferenčného systému tak, že sme poskytli detailný zoznam požiadaviek. To nám umožní vybrať vhodné protokoly, kodeky a komponenty. Z voľne dostupných technológií vyberáme AccessGrid, pretože vo svete open-source sa jedná o najkomplexnejší videokonferenčný systém, ktorého základ využívajú aj niektoré komerčné produkty.

Chapter 3

Protocols

3.1 Multicast

Vzhľadom na skutočnosť, že v rámci videokonferencie spolu komunikuje istá skupina uzlov, začneme túto kapitolu vysvetlením rozdielov medzi spôsobmi doručovania paketov v kontexte skupinovej komunikácie. Vysvetlíme rozdiely medzi stratégiami **unicast** a **multicast**¹. Z pohľadu spoľahlivosti nám pôjde o nespoľahlivé a nespojované služby, ktoré sú vhodné pre multimedialne prenosy. Zabezpečenie spoľahlivého doručovania by malo za následok nárast latencie. Rozdiel v efektivite doručovania sme schematicky znázornili na obrázku 3.1. Je dôležité si uvedomiť, že popisujeme situáciu, kedy jeden odosielateľ posiela rovnakú správu skupine príjemcov. Nielenže chceme riešiť otázku **efektivity**, ale aj otázku **adresovania skupiny** príjemcov.

Unicast: Je spôsob doručovania, kde odosielateľ musí každému príjemcovi osobitne zaslať každý paket. Je to základný spôsob doručovania v súčasnom internete, ktorý však plytvá prenosovou kapacitou siete, pretože vyžaduje duplikovanie pake-

¹Doručovanie formou broadcastu nás zaujímať nebude, pretože sa väčšinou obmedzuje na doručovanie v rámci jedného segmentu siete. Navyše multicast je silnejší pojem a dokáže ho efektívne emulovať.

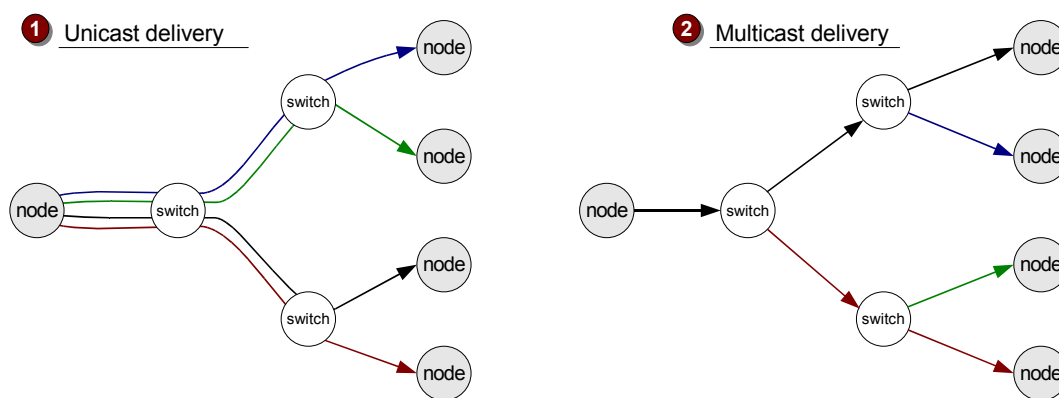


Figure 3.1: Rozdiely medzi doručovaním metódou unicast a multicast.

tov. Navyše odosielateľ musí poznať adresu každého príjemcu. Na obrázku 3.1 (1) je táto situácia znázornená ako 4 dátové toky vychádzajúce od odosielateľa.

Multicast: Je spôsob doručovania, kedy odosielateľ posiela pakety na skupinovú adresu. Pakety sú duplikované až pri prechode medzi siet'ami. Multicast tak šetrí prenosovú kapacitu. Pre správne fungovanie musia všetky uzly v sieti vrátane prepínačov a smerovačov podporovať tento spôsob doručovania, čo vyžaduje úpravy v pôvodnom unicastovom spôsobe doručovania (RFC 1112 [11]). V protokole IPv4 je pre multicast vyhradená skupina adries tzv. class D. Sú to adresy začínajúce v binárnom zápise prefixom 1110, v protokole IPv6 prefixom 11111111. Každá adresa určuje multicastovú skupinu pre doručovanie. Členstvo v týchto skupinách riadi protokol IGMP. Skupina zariadení, ktoré podporujú protokol IGMP, tvorí v rámci internetu virtuálnu sieť MBone (multicast backbone). Sieť je tvorená buď špecializovanými hardwarovými komponentmi, alebo počítačmi s podporou IGMP v operačnom systéme. Smerovanie multicastových paketov navyše vyžaduje podporu niektorého z multicastových routovacích protokolov DVMRP (multicastová verzia RIP), MOSPF (multicastová verzia OSPF) alebo PIM (Protocol Independent Multicasting). Táto práca však o routovaní nepojednáva.

V súčasnosti je dostupnosť multicastových služieb na nízkej úrovni a pre bežných užívateľov prakticky nedostupná. Zmenu by mal priniesť prechod na protokol IPv6, ktorý obsahuje multicast ako svoju neoddeliteľnú súčasť. Každopádne, univerzity a iné vybrané inštitúcie majú k multicastu väčšinou prístup.

3.1.1 IGMP

Pre pochopenie multicastových služieb v IPv4 sieti je nutné poznať protokol IGMP – Internet Group Management Protocol, ktorý pomáha udržiavať informácie o zložení skupín na danom sieťovom segmente. Pre správne doručovanie multicastových paketov musí byť podporovaný všetkými koncovými uzlami a smerovačmi. (Smerovače môžu byť totiž členmi multicastovej skupiny).

Táto informácia je dôležitá hlavne pri nákupe potrebného hardware. Smerovače s podporou IGMP sú totiž podstatne drahšie ako ich bežné ekvivalenty. V dobe tvorby tohto textu sa cena najlacnejšieho zariadenia² pohybovala okolo 5000 CZK bez DPH. Priemerná cena je však okolo 10000 CZK.

4b	1b	3b	8b	16b
Version	Type	Unused	Max. response time	Checksum
Group address				
IGMP packet structure				
⋮				

Table 3.1: Formát IGMP paketu.

Multicastový smerovač na každom segmente vysiela periodické správy Membership Query (tab. 3.2), kde Group Address = 0 (tzv. General Membership Query).

²Allied Telesyn AT-AR300V2SECPK

Uzol, ktorý je členom tejto skupiny, odpovie zaslaním správy typu Membership Report (tab. 3.3). Aby nedochádzalo k zahľteniu segmentu odpovedí od všetkých uzlov v skupine, každý uzol čaká náhodne zvolený interval pred tým, než začne vysielat'.

Uzol, ktorý sa prihlasuje alebo opúšťa skupinu, zasiela tzv. State Change Report, čo je modifikovaný Membership Report (tab. 3.3). Pri odchode uzla zo skupiny môže nastať situácia, že skupina už neobsahuje ďalšie uzly. V takom prípade by mala skupina zaniknúť. Po každom odchode nejakého uzla odošle smerovač Group Specific Membership Query (znova modifikovaná verzia Membership Query) a uzly odpovedajú znova pomocou Membership Report.

Detailný popis protokolu IGMP sa nachádza v RFC 3376 [6].

4b	1b	3b	8b	16b
Type = 0x11			Max Resp Code	Checksum
Group address				
Resv	S	QRV	QQIC	Number of Sources (N)
Source Address [1]				
⋮				
Source Address [N]				

Table 3.2: IGMP Membership Query.

8b	8b	16b
Type = 0x22	Reserved	Checksum
Reserved		Number of Group Records (N)
Group Record [1]		
⋮		
Group Record [N]		

Group Record má nasledujúci formát:

8b	8b	16b
Record Type	Aux Data Len	Number of Sources (N)
Multicast Address		
Source Address [1]		
⋮		
Source Address [N]		
Auxiliary Data		

Table 3.3: IGMP Membership Report.

3.1.2 SAP

Session Announcement Protocol (SAP), definovaný v RFC 2327 [22], umožňuje periodicky vysielat' informácie o prebiehajúcom multimedialnom prenose na vybrané

multicastové adresy. Potenciálni príjemcovia SAP paketu môžu tento popis použiť na spustenie a nastavenie potrebného dekódovacieho nástroja.

3b	1b	1b	1b	1b	1b	8b	16b
V = 1	A	R	T	E	C	auth len	msg id hash
originating source (32 or 128 bits)							
optional authentication data							
⋮							
optional payload type							
...	0	...	payload				

Table 3.4: Formát SAP paketu.

Príklad: Nech existuje odosielateľ, ktorý generuje RTP pakety na multicastovú adresu 224.0.0.1. Obraz zakódovaný kodekom H.263 odosiela na port 10000, zvuk zakódovaný ako L16 na port 20000. Pre príjem obrazu a zvuku je preto nutné spustiť na strane príjemcu aplikácie VIC a RAT:
 vic 224.0.0.1/10000 & rat 224.0.0.1/20000

3.1.3 MADCAP

8b	8b	16b
version	msgtype	addrfamily
xid		
Options		
⋮		

Table 3.5: Formát MADCAP paketu.

Tento protokol podobný DHCP umožňuje dynamicky alokovať multicastové adresy. Funguje ako alternatíva k statickej alokácii, kde adresy priradzuje administrátor. Podľa modelu ISO/OSI hovoríme o protokole aplikačnej vrstvy. Používa port 2535 (UDP). Klienti kontaktujú niektorý z MADCAP serverov pomocou unicastu alebo multicastu. Server odpovedá vždy pomocou unicastu. Protokoly DHCP a MADCAP sa používajú v disjunktných sférach.

3.1.4 UMTF

Riešenie problémov spojených s nedostupnosťou multicastových služieb by mal riešiť protokol UMTF [14], ktorého štandardizácia je v stave IETF Internet-Draft. Ako ukazujú výsledky z najnovšej verzie AccessGridu 3, môžeme sa tešiť na silný nástroj, ktorý rieši problémy spojené s prechodom cez NAT a firewall. Jednou z prvých implementácií je aplikácia AGConnector, ktorej fungovanie znázorňuje obrázok 7.7.

3.2 Multimedia-related protocols

3.2.1 RTP and RTCP

RTP (Real Time Transport Protocol) je protokol navrhnutý pre prenos dát v reálnom čase. Podľa modelu ISO/OSI ide o protokol aplikačnej vrstvy. Spolu s protokolom RTCP (Real Time Control Protocol) tvorí rámec vhodný hlavne na prenos multimediálnych dát prostredníctvom unicastovej alebo multicastovej siete.

Zatiaľ čo RTP prenáša samotné dáta, RTCP umožňuje monitorovať a riadiť doručovanie RTP paketov. Protokoly neobsahujú funkcie pre rezerváciu zdrojov ani QoS (quality of service), pre ktoré je potrebná spolupráca s nižšími vrstvami³. Umožňujú ale identifikovať možné zmeny v sieti a prispôbiť vlastnosti vysielateľa alebo prijímateľa napr. zmenou kodeku, kvality alebo bufferovania. Oba protokoly sú navrhnuté tak, aby boli nezávislé na použitých sieťových a transportných vrstvách.

Detailný popis oboch protokolov sa nachádza v RFC 3550 [53]. Pre naše účely je vhodné spomenúť, že pri videokonferenčných prenosoch bude zvuk a obraz prenášaný nezávislými dátovými tokmi. Každý tok je reprezentovaný dvoma UDP portami – obvykle port s párnym číslom N pre RTP-data a port $N+1$ pre RTCP pakety. Tvorcovia protokolu vychádzali z predpokladu, že niektorí účastníci môžu požadovať iba príjem niektorých dátových tokov (napr. iba príjem zvuku z dôvodu pomalejšieho linky). Takéto rozdelenie však v praxi spôsobuje problémy pri prechode cez NAT/firewall. Úspešnejší sa zdá byť protokol IAX2 (kapitola 3.3.3), ktorý prenáša všetky dáta na jednom UDP porte.

RFC 3550 popisuje aj využitie RTP mixerov, ktoré umožňujú multiplexing dátových tokov a RTP translatorov, meniacich obsah dátového toku napr. pri preklade medzi multicastovou a unicastovou sieťou.

Synchronizácia dátových tokov je zabezpečená časovými razítkami obsiahnutými v rámci RTCP.

0b	0b	0b	0b	0b	0b	0b
V=2	P	X	CSRC count	M	payload type	sequence number
timestamp						
synchronization source (SSRC) identifier						
contributing source (CSRC) identifiers						
⋮						

Table 3.6: Formát RTP paketu.

Tabuľka 3.6 popisuje formát RTP paketu, z ktorého sú pre nás najdôležitejšie nasledujúce položky:

payload type: Informácia o obsahu paketu, ktorá je buď implicitne definovaná v RFC 3551 [52], alebo dohodnutá mimo RTP.

sequence number: Číslo inkrementované s každým paketom umožňuje detekciu straty paketov. Na začiatku session býva obvykle zvolená náhodná hodnota.

³Pre rezerváciu zdrojov a QoS je možné použiť protokol RSVP popísaný v kapitole 3.2.2.

timestamp: časové razítko dátového toku, závisí na obsahu, umožňuje výpočet hodnoty jitter (rozptyl reakčnej odozvy). Na začiatku session je hodnota obvykle volená náhodne a inkrementovaná podľa vnútorných hodín odosielateľa.

synchronization source (SSRC) identifier: Umožňuje príjemcovi identifikovať entitu, ktorá paket odoslala.

3.2.2 RSVP

Resource Reservation Protocol (RSVP) [5] umožňuje príjemcovi požadovať rezerváciu sieťových zdrojov počas unicastových alebo multicastových dátových prenosov. Pre správne fungovanie QoS musia všetky smerovače na ceste od zdroja k príjemcovi podporovať RSVP. Podľa modelu ISO/OSI ide o protokol transportnej vrstvy, ktorý pracuje nad IPv4 alebo IPv6. RSVP neprenáša aplikačné dáta, funguje len ako riadiaci mechanizmus. RSVP sa zaoberá iba nastavením QoS pre prenášané pakety na rozdiel od smerovacích protokolov, ktoré rozhodujú o ceste paketov.

4b	8b	16b	32b
V=2	flags	Message Type	RSVP Checksum
Send TTL	Reserved		RSVP Length
RSVP header structure			

Table 3.7: Formát RSVP paketu.

3.2.3 RTSP

Real Time Streaming Protocol (RTSP) [54] môžeme podľa modelu ISO/OSI zaradiť medzi protokoly aplikačnej vrstvy. Využíva sa pri streamovaní multimediálnych dát. Umožňuje prijímajúcej strane ovládať priebeh streamovania. Syntax pripomína HTTP a podporovanými metódami sú napr. PLAY, RECORD, SETUP, PAUSE atď. ... Protokol RTSP nie je určený na prenos dát, ten je riešený iným mechanizmom, napr. pomocou RTP. Môžeme ho nájsť implementovaný napr. v aplikácii MPlayer ako jeden z možných spôsobov prenosu multimediálneho obsahu.

Príklad: `mplayer rtsp://path/to/stream`

3.2.4 SDP

Session Description Protocol (SDP), popísaný v RFC 2327 [21], je vhodný na popis multimediálnych dátových tokov v unicastovej alebo multicastovej sieti. Umožňuje nastaviť parametre aplikácie, ktorá bude tieto dáta interpretovať. Podľa modelu ISO/OSI ide o protokol aplikačnej vrstvy.

Príklad: Streamovací server na adrese 239.255.42.42 má uložené prednášky, ktoré môže klient prijať vo vysokej alebo v nízkej kvalite. Každý takýto dátový tok je reprezentovaný SDP súborom. Pre príjem môže použiť napr. aplikáciu MPlayer:

```
mpplayer sdp://presentation_01_low.sdp
```

...Príslušný SDP súbor môže vyzerat' nasledovne:

```
v=0
o=- 49451 3 IN IP4 127.0.0.1
s=Test MPEG Audio+Video session
i=Session streamed by testMPEG1or2AudioVideoStreamer
t=0 0
a=tool:testMPEG1or2AudioVideoStreamer
a=type:broadcast
m=audio 6666 RTP/AVP 14
c=IN IP4 239.255.42.42/127
m=video 8888 RTP/AVP 32
c=IN IP4 239.255.42.42/127
```

3.3 VoIP protocols

3.3.1 SIP

Session Initiation Protocol je otvorený signalizačný protokol určený pre uskutočňovanie telefónnych hovorov prostredníctvom IP siete. Jedná sa o textovo-orientovaný protokol aplikačnej vrstvy, je definovaný v RFC 3261 [49]. Spojenie je zabezpečené prostredníctvom dvoch typov uzlov – User Agent a Server. Spojenie môže byť naviazané aj medzi dvoma klientmi bez účasti servera. Niekoľko aplikácií, používajúcich SIP, sme popísali v kapitole 2.3.1. Užívateľ je adresovaný pomocou URL `sip:username@hostname`. Prechod cez firewall a NAT je zabezpečený pomocou STUN servera.

3.3.2 H.323

Protokol z dielne ITU, ktorý konkuruje protokolu SIP. Je integrovaný v niektorých hardwarových zariadeniach určených pre IP telefóniu. Túto kapitolu uvádzame iba z didaktických dôvodov, pretože na niekoľkých miestach v texte práce spomíname zariadenia podporujúce H.323. Napríklad systém VRVS disponuje premostením zo sveta H.323.

3.3.3 IAX2

Inter Asterisk Exchange je binárny protokol určený pre komunikáciu telefónnych ústrední Asterisk⁴ a ďalšie VoIP služby. Jeho štandardizácia je zatiaľ v štádiu IETF Internet-Draft [56]. Podľa modelu ISO/OSI je IAX protokolom aplikačnej vrstvy. Využitie by mohol nájsť aj vo veľkých videokonferenčných systémoch akým je AccessGrid, prípadne ako protokol pre streamovanie multimediálneho obsahu. Kombinuje v sebe vlastnosti **kontrolného**, **transportného** a **signalizačného** protokolu, takže zastupuje funkčnosť protokolov RTP/RTCP, SIP, H.323. Na prenos

⁴Asterisk PBX uvádzame v kapitole 2.3.1

všetkých IAX paketov sa využíva jeden statický UDP port 4569, to mu umožňuje prekonávať problémy spojené s firewallmi a NAT. IAX nepotrebuje externý proxy server ako napríklad SIP, ani žiadne špeciálne nastavenie firewallu (okrem neblokovania portu 4569). Môžeme preto tvrdiť, že by mohol nahradiť protokoly RTP/RTCP, SIP, H.323.

3.4 Evaluation of selection of protocols

Spomínané protokoly pokrývali tri dôležité oblasti.

Prvá skupina sa týkala multicastu, takže bolo potrebné spomenúť najdôležitejší protokol IGMP, bez ktorého multicast nefunguje. Protokolom MADCAP sme chceli poukázať na možnosti automatickej konfigurácie multicastových skupín. V budúcnosti by to mohla byť cesta ako vnieť poriadok do inak chaotického sveta náhodne volených multicastových adries. Navyše nám umožňuje dosiahnuť rovnaké pokroky ako DHCP vo svete statických IP adries. Protokol SAP zas umožňuje popísať už existujúci multicastový prenos.

Druhá skupina sa týkala transportu a riadenia toku multimediálnych dát. Najdôležitejší protokol RTP, spolu s jeho partnerom RTCP, sme popísali podrobnejšie, pretože táto téma súvisí s informáciami v ďalšom texte. RTSP a SDP sme spomenuli v súvislosti so streamovaním multimediálneho obsahu. Ukázali sme aj krátky príklad prehrávania pomocou aplikácie MPlayer. Protokol RSVP spomíname pre jeho QoS vlastnosti.

Poslednou popisovanou skupinou boli protokoly, ktoré sú využívané VoIP aplikáciami. Najzaujímavejším sa nám javí protokol IAX2, najmä jeho schopnosť bezproblémového prechodu cez NAT/firewall a schopnosť nahradiť niekoľko vyššie popisovaných protokolov. Protokol SIP sme spomínali v súvislosti so STUN serverom a ako zástupcu skupiny signalizačných protokolov.

□

Chapter 4

Codecs

Kodek je softwarový komponent poskytujúci dve najdôležitejšie funkcie pre prácu so signálom – kódovanie a dekódovanie. **Kodér** zabezpečuje prevod analógového resp. digitálneho obrazu v pôvodnom formáte (raw format) do formátu s inými vhodnými parametrami (encoded format), v prostredí počítačových sietí je dôležitým parametrom veľkosť prenášaných dát. Zníženie veľkosti sa dosahuje za cenu nižšej kvality a nárastu latencie resp. jitteru počas kódovania. Prevod kódovaných dát naspäť do pôvodného tvaru má za úlohu komponent **dekodér**. V prípade videokonferencie sa proces kódovania a dekódovania dotýka signálov analógovej povahy, teda zvuku a obrazu. Preto budeme hovoriť o tzv. video a audio kodekoch.

4.1 Need of codecs

Ukážeme, prečo je vôbec nutné používať kódovanie obrazu. Prečo nestačí používať signál v originálnej podobe. (Podobný príklad je možné uviesť aj v prípade prenosu audio signálu.)

Obraz v reálnom svete je snímaný niekoľkokrát za sekundu a rozdelený do snímok (frames). Túto hodnotu označíme *fps* (frames per second). Každý frame sa skladá z bodov (pixels), kde každý pixel je reprezentovaný ako n -tica bitov a táto kombinácia predstavuje istú farbu vo farebnom priestore RGB.

Príklad: Pre jednoduchosť predpokladajme, že máme obraz v rozlíšení CIF 352x288 pixelov pri použití 24-bitovej farebnej hĺbky s frekvenciou 20fps. (24bit RGB predstavuje : $256 = 2^8$ hodnôt červenej farby, $256 = 2^8$ hodnôt zelenej farby, $256 = 2^8$ hodnôt modrej farby = $2^{24} = 16777216$ rôznych farieb.) Výsledný dátový tok pre jeden takýto signál by dosiahol:

$$\text{width} \times \text{height} \times \text{depth} \times \text{fps} = 352 \times 288 \times 24 \times 20 = 48660480 \text{ b/s} = \mathbf{5.7 \text{ MiB/s}}$$

Táto hodnota je ďaleko za možnosťami súčasných sietí, keď si predstavíme, že v rámci konferencie sa musí prenášať súčasne niekoľko obrazových tokov. V prípade veľkosti D1 PAL táto hodnota vychádza dokonca **23 MiB/s**.

4.2 Video and audio transmission using RTP protocol

V nasledujúcej časti uvedieme niektoré vybrané kodeky využívané dnešnými videokonferenčnými nástrojmi. Väčšina týchto systémov používa na prenos protokol RTP/RTCP, čo špecifikuje RFC 3551 [52]. V prípade obrazu sú to kodeky: CelB, JPEG (Motion JPEG), H.261, H.263-1998 (H.263+), MPV, MP2T, nv. V prípade zvuku sú to kodeky: DVI4, G722, G723, G726-40, G726-32, G726-24, G726-16, G728, G729, G729D and G729E, GSM, GSM-EFR, L8, L16, LPC, MPA, PCMA, PCMU, QCELP, RED a VDWI.

Ďalším dôležitým dokumentom je RFC 2198 [45], kde je popísaný spôsob prenosu audio paketov v sieťach, kde môže dochádzať k ich strate. Aby sa zabránilo výpadkom zvuku, používa sa niekoľko redundantných kódov okrem hlavného kódovania.

4.2.1 Video: NV

Historický video-kodek od Xerox PARC implementovaný napr. v aplikácii VIC z dielne UCL.

4.2.2 Video: H.261

V roku 1990 bol štandardizovaný dodnes používaný nízkolatenčný kodek H.261, navrhnutý pre prenos obrazu veľkosti QCIF a CIF (obrázok 2.7). Na obrázku 4.1 je znázornený lepší pomer kvality k dátovému toku oproti staršiemu kodeku NV. Podporu H.261 nájdeme v širokom spektre aplikácií (vic, mplayer, vlc, ffmpeg), hlavne z dôvodu kompatibility so staršími systémami. Prenosom obrazu kódovaného pomocou H.261 sa zaoberá aj RFC 2032 [61].

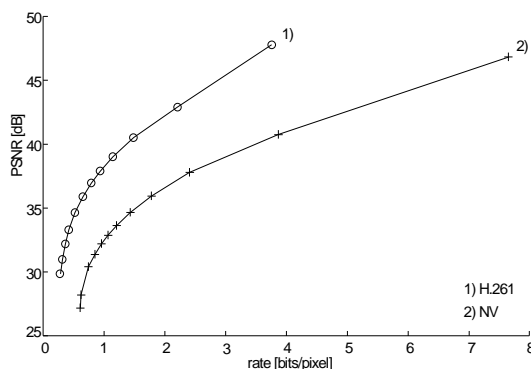


Figure 4.1: Porovnanie kodeku H261 a NV. (zdroj [42])

4.2.3 Video: H.263

Kodek H.263 je prirodzeným následníkom H.261. Štandardom sa stal v roku 1996. Kombinuje znaky z H.261 spolu s technológiou MPEG a optimalizuje ich pre nízke dátové toky. V porovnaní s jeho starším kolegom je oveľa efektívnejší pri všetkých dátových tokoch, ako vidíme na obrázku 4.2, navyše podporuje viac veľkostí obrazu: Sub-QCIF, QCIF, CIF, 4CIF, 16CIF. Prenosom obrazu kódovaného pomocou H.263 sa zaoberá RFC 2190 [64].

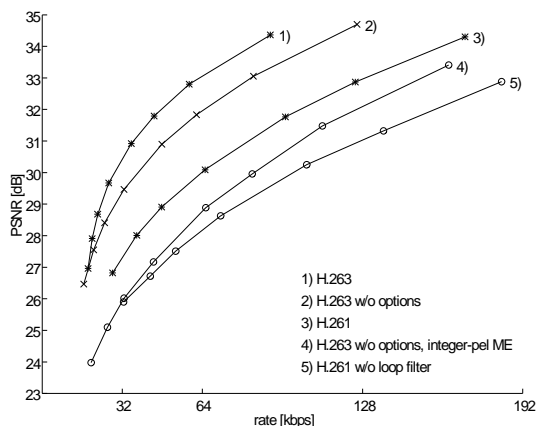


Figure 4.2: Porovnanie kodeku H261 a H263. (zdroj [19])

4.2.4 Video: H.264 (MPEG-4 AVC)

Jeden z najnovších kodekov určený pre videokonferenčné účely, ktorý by mal nahradiť kodek H.263, čo znázorňuje obrázok 4.3. Je to štandard, ktorý by mal zastrešovať kódovanie obrazu pre účely interaktívnej komunikácie, streamovania, a broadcastu. Najväčším problémom pri rozširovaní tohto kodeku sú jeho vyššie nároky na procesor, kde sa uvádza až 8-krát zložitejšie kódovanie a 3-krát zložitejšie dekódovanie oproti MPEG-2 (zdroj [20]). Prenosom H.264 pomocou protokolu RTP sa zaoberá RFC 3984 [62].

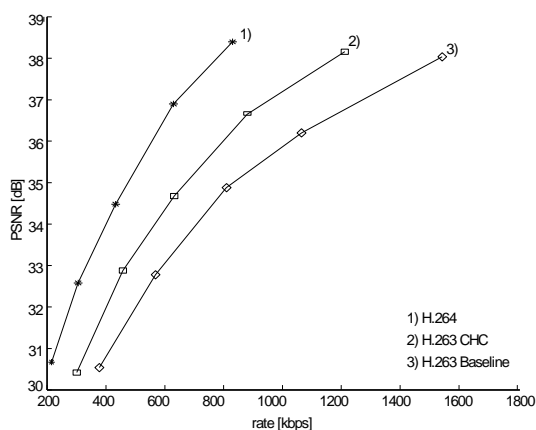


Figure 4.3: Porovnanie kodeku H263 a H264. (zdroj [30])

4.2.5 Video: MPEG-2

Tento kodek umožňuje komunikáciu vo vysokej kvalite pri šírke pásma 1.5Mbit/s až 15Mbit/s. Problémy s kompatibilitou medzi rôznymi implementáciami však zabraňujú jeho masovému rozšíreniu. Pokiaľ hovoríme o kvalite kódovania, vykazuje horšie parametre ako H.264 (obrázok 4.4). Na druhej strane je však menej náročný na procesor.

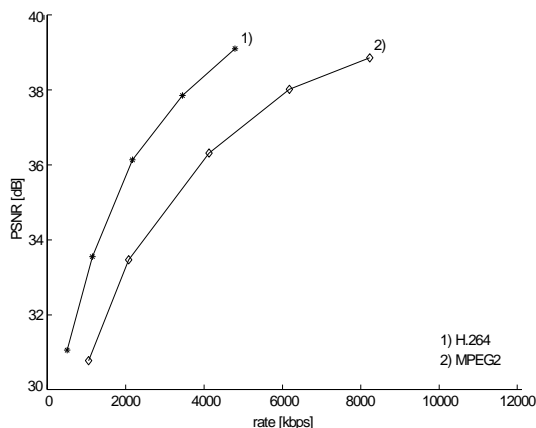


Figure 4.4: Porovnanie kodeku H264 a MPEG-2. (zdroj [30])

4.2.6 Video: Motion JPEG

MJPEG kóduje obrazové snímky ako osobitné JPEG obrázky, ktoré sú pri prenose rozdelené do RTP paketov (RFC 2435 [4]). Výhodou je rýchlosť kompresie a dekompresie ako aj možnosť úpravy obrazového toku bez dopočítavania prechodov medzi jednotlivými snímkami. Táto jeho vlastnosť sa využíva pri úpravách obrazu v reálnom čase. MJPEG navyše umožňuje posielat' prakticky ľubovoľné veľkosti obrazu vo vysokej kvalite, čo sa využíva napr. pri posielaní snímkov pracovnej plochy. Nevýhodami sú vyššie nároky na prenosovú kapacitu a väčšia náchylnosť k chybám spôsobených nekvalitným spojením a stratou RTP paketov.

4.2.7 Audio: L16

Jednoduchý audio-kodek, ktorý umožňuje prenos nekomprimovaného zvukového signálu v rozlíšení 16 bitov. Úroveň signálu je rozložená na hodnoty -32768 až 32767. Používa sa ako implicitný kodek v aplikácii RAT (kapitola 7.5.3). Jedna z verzií tohto kodeku (16KHz mono) je kompatibilná so štandardom G.711 [28]. Dokument RFC 3551 [52] špecifikuje RTP Payload profile pre L16 Audio 44100Hz 1/2ch.

4.2.8 Audio: GSM

GSM (Groupe Speciale Mobile) definuje EU standard GSM 06.10 pre kódovanie hlasu s dátovým tokom 13,200 b/s. Dokument RFC 3551 [52] špecifikuje RTP Payload profile pre GSM Audio 8000Hz 1ch.

4.2.9 Audio: G.722

Tento proprietárny kodek sa používa v hardwarových H.323 klientoch, špecifikovaný v ITU-T G.722 [27].

4.2.10 Audio: Speex

Kodek vznikol ako otvorený štandard a nie je zat'azený softwarovými patentami. Zatiaľ ešte nevzniklo RFC pre RTP Payload Format kodeku Speex, ale existuje draft

Codec	B/ frm	frm (ms)	frm/ pak	pak/ sec	pay (B)	B/ pak	bits/ sec	latency
G723.1	20	30	1	33.33	20	60	16000	30
G723.1	24	30	1	33.33	24	64	17067	30
GSM	33	20	1	50	33	73	29200	20
GSM	33	20	2	25	66	106	21200	40
G.711	240	30	1	33.33	240	280	74667	30
LPC10	7	22.5	1	44.44	7	47	16711	23
LPC10	7	22.5	2	22.22	14	54	9600	45
Speex-5.95	15	20	1	50	15	55	22000	20
Speex-5.95	15	20	2	25	30	70	14000	40
Speex-8k	20	20	1	50	20	60	24000	20
Speex-8k	20	20	2	25	40	80	16000	40
Speex-11k	28	20	1	50	28	68	27200	20
Speex-11k	28	20	2	25	56	96	19200	40
Speex-15k	38	20	1	50	38	78	31200	20
Speex-15k	38	20	2	25	76	116	23200	40
Speex-18.2	46	20	1	50	46	86	34400	20
Speex-18.2	46	20	2	25	92	132	26400	40
Speex-24.6	62	20	1	50	62	102	40800	20
Speex-24.6	62	20	2	25	124	164	32800	40

Table 4.1: Audio kodeky s latenciou maximálne 50ms. (zdroj [55])

dokument [23], ktorý sa touto problematikou zaoberá. Speex má šancu stať sa najpoužívanejším kodekom vo videokonferenčných aplikáciách s otvoreným zdrojovým kódom, pre jeho voľbu hovoria parametre uvedené v literatúre [55].

4.3 Evaluation of selection of codecs

Popísali sme niekoľko bežne používaných kodekov, ktoré sú vhodné pre videokonferenčné účely. Uviedli sme dôvody, prečo je počas sieťového prenosu vôbec nutné uvažovať nad kódovaním multimediálnych dát. Z video-kodekov sme vybrali ako najvhodnejšieho kandidáta kodek H.263 pre jeho rozšírenie v existujúcich systémoch a kodek H.264 pre jeho výborný kompresný pomer. Z audio-kodekov dávame svoje preferencie kodeku Speex. Vzhľadom na využitie G722 a H.261 v mnohých hardwarových H.323 klientoch je vhodné uvažovať aj o tejto podpore.

Na záver kapitoly by sme ešte radi uviedli tabuľku 4.1, ktorá obsahuje audio-kodeky s latenciou maximálne 50ms. Ide o nízkolatenčné kodeky, ktoré môžeme využiť pri spolupráci na projekte s MARCEL network (kapitola 5.2).

□

Chapter 5

CIANT AccessGrid node

Jedným z hlavných výsledkov tejto diplomovej práce je videokonferenčný uzol v neziskovej organizácii CIANT, ktorý môže inšpirovať aj ďalšie inštitúcie v snahe získať videokonferenčnú technológiu.

CIANT (International Centre for Art and New Technologies) je nevládna nezisková organizácia založená v roku 1998. Pôsobí na mezinárodnej úrovni v kultúrnej, vzdelávacej a výkumnej oblasti v spojení s novými technológiami. Prevádzkuje multimedialne laboratórium so zameraním na medziodborovú spoluprácu. Práve orientácia na nové technológie umožnila vznik jedného z prvých AccessGrid uzlov v Českej republike. Významnú podporu v tomto projekte poskytla Česká akademická sieť CESNET vo forme multicastového pripojenia s rýchlosťou 15 Mbit/sec. Odborné konzultácie poskytli v úvode projektu Petr Holub, Masarykova Univerzita v Brne a Grzesiek Sedek, Wimbledon School of Art.

Naším pôvodným cieľom nebolo vytvoriť plnohodnotný videokonferenčný uzol. Chceli sme najskôr preskúmať možnosti ponúkaného riešenia vo forme AccessGridu. Postupne sme však prispôbili jednu z miestností pre videokonferenčné účely a okrem pasívnej účasti sme nakoniec prešli až k implementácii unicast-to-multicast mostu a sprístupnili sme tak pripojenie aj pre užívateľov bez multicastového pripojenia.

Za hlavný úspech považujeme spoluprácu s ostatnými AccessGrid uzlami zo skupiny MARCEL network, ktorá združuje umelcov, vedcov a inštitúcie so záujmom o spoluprácu prostredníctvom vysokorýchlostnej siete. Cieľom je podpora interdisciplinárnej komunikácie vo vedeckých a umeleckých projektoch. Typickým príkladom je cyklus konferencií **Polylogues**, ktorému je venovaná kapitola 5.1. MARCEL má v dnešnej dobe vyše 100 členov z viac ako 17 krajín a jedným z členov sa najnovšie stala aj Karlova Univerzita.

5.1 Series of Polylogue video-conferences

Uzol AccessGridu sa vyvíja od roku 2005 a je zaregistrovaný v zozname AGCentral [66]. Má za sebou niekoľko úspešných videokonferencií organizovaných v rámci projektu **Polylogues 2006**, kde sa uplatnila spolupráca s Filozofickou fakultou Univerzity Karlovej, katedrou nových médií a MARCEL network.

Museums of the Future[32]: Prvý videokonferenčný prenos z cyklu Polylogues sa konala 15.2.2006. Okrem uzla v Ciante sa na ňom podieľali uzly z USA – University of Maine, Georgia Institute of Technology, uzol z Veľkej Británie – University of East London a prvý AccessGridový uzol z Českej republiky – ANTLab, Masarykova Univerzita v Brne.

Game Engines/Social Engines[31]: S mesačným odstupom nasledoval druhý prenos 29.3.2006, kde sa zúčastnili uzly Georgia Institute of Technology, University of East London a Masarykova Univerzita v Brne.

Visualizations in Art and Science[33]: Posledný prenos z cyklu Polylogues sa konal 24.5.2006 za účasti University of East London a Masarykovej Univerzity v Brne.

5.2 AccessGrid in the art community

Ukazuje sa, že technológia AccessGrid, pôvodne určená pre spoluprácu vedeckých tímov, vyžaduje v kontexte kultúry a umenia nástroje s novými vlastnosťami. V rámci združenia MARCEL sa preto pripravuje projekt s pracovným názvom **Artists' needs for a multicasting platform**, ktorého hlavným cieľom je umožniť umeleckej komunite efektívne využívať priestor vysokorýchlostných sietí. Umenie akým je tanec, hudba, divadlo a iná interaktívna komunikácia medzi umelcami a ich publikom vyžaduje niekoľko výrazných zmien v súčasnej architektúre AccessGridu. Vznikol tak zoznam nasledujúcich požiadaviek:

1. Minimalizácia **latencie**, veľmi striktný limit 50 milisekúnd. Táto požiadavka ovplyvní hlavne výber **kodeku** a **protokolov** určených na prenos multimediálneho obsahu.
2. Dôležitým faktorom pri komunikácii v umení je **priamy očný kontakt**, ktorý vyžaduje správnu polohu a ovládanie kamery.
3. Nový systém musí viac vychádzať v ústrety technicky menej skúsenej časti populácie. Táto požiadavka si vyžiada nielen zjednodušenie samotného AG nástroja, ale aj vytvorenie nových aplikácií založených na webových technológiách dostupných kedykoľvek z internetového prehliadača. Účastníci by mali dostať nástroj umožňujúci lepšiu orientáciu v priestore virtuálnych miestností a udalostí.
4. Rozšírené možnosti monitorovania technického stavu celého systému. Táto funkcia ale musí byť skrytá pred bežnými užívateľmi pre udržanie dostatočnej miery jednoduchosti.
5. Osvetlenie divadelnej scény má byť závislé na podmienkach sieťového prenosu.
6. Manipulácia s obrazom v reálnom čase, aplikovanie špeciálnych efektov.
7. Možnosť vizuálne zväčšiť rozmery miestnosti napríklad pomocou veľkoplošnej projekcie na zadnú stenu.

8. Lepšie možnosti nahrávania videokonferenčného prenosu s možnosťou simultánne nahrávať dianie vo všetkých uzloch.
9. Interaktivita s publikom prostredníctvom webu a distribúcia multimedialného obsahu vo forme video-on-demand.

Pri návrhu a implementácii častí požiadaviek sa môžeme inšpirovať existujúcimi riešeniami ako napr. VRVS/EVO, či InSORS. Požiadavky číslo 2, 5, 6, 7 a 8 sú však unikátne a nie sú zatiaľ implementované žiadnym videokonferenčným systémom spomínaným v kapitole 2. Systém VRVS môže ponúknuť vzor pre spôsoby monitorovania infraštruktúry pomocou distribuovanej siete MonaLisa. Navyše pre bežného užívateľa je práca s VRVS-rozhraním jednoduchšia už aj preto, že aplikáciu je možné spustiť priamo z internetového prehliadača. Systém reflektorov a zabudovaný nástroj RTP-proxy rieši niektoré problémy spojené s NAT/firewallom.

Požiadavka minimálnej latencie v 1. bode obmedzuje použité zvukové kodeky na množinu uvedenú v tabuľke 4.1. Ako najperspektívnejší sa javí open-source kodek Speex. Súčasná verzia aplikácie RAT však tento kodek zatiaľ nepodporuje.

5.3 Information portal

Dôležitou súčasťou projektu bola myšlienka informačného portálu, ktorý by obsahoval všetky informácie o plánovaných i archivovaných konferenciách, rôzne návody a popis nastavenia AG a zároveň by umožnil lepšie zdieľanie informácií v rámci organizácie. Pre tento účel sa osvedčil jednoduchý systém Wiki, ktorý s odstupom času hodnotíme ako vysoko produktívny. Preto by sme ho radi doporučili každej organizácii, ktorá uvažuje o vytvorení videokonferenčného uzla.

Pri výbere sme porovnávali množstvo existujúcich wiki enginov, nakoniec sa však jednoduché riešenie ukázalo ako najpoužiteľnejšie. Rozhodli sme sa pre engine PmWiki. Presvedčili nás tieto vlastnosti:

- PmWiki ukladá stránky do súborov narozdiel napr. od MediaWiki, ktorá používa relačnú databázu. Aj napriek tomu je vyhľadávanie v relatívne veľkom množstve stránok dostatočne rýchle. Odpadá tak nutnosť inštalácie a správy databázy.
- Veľmi jednoduchá inštalácia – stačí rozbalit' archív, nastaviť heslá pre prístup k stránkam a zmeniť logo.
- PmWiki umožňuje nastaviť globálne heslo pre editovanie, upload, prezeranie stránok, ako aj heslá špecifické pre skupiny stránok a dokonca aj pre samostatné stránky.
- Základný engine je možné obohatiť o ďalšie moduly (plugins).

PmWiki bolo nutné pred použitím značne upraviť. Môžeme spomenúť napríklad implementáciu kódovania UTF-8, podporu PHP5, prispôbenie vzhľadu tak, aby uľahčoval orientáciu a úpravu stránok, hlavne prácu s obrázkami a sekciami. Pridaných bolo aj niekoľko bezpečnostných prvkov, hlavne podporu pre SSL pri editácii stránok a pri práci s heslami. Takto upravenú verziu je možné stiahnuť na stránkach projektu [68]. Užívateľ dostane do rúk funkčné a odskúšané riešenie.

5.4 Components used during the node setup

Pri voľbe komponentov uzla sme kládli dôraz na minimalizáciu finančných nákladov. Snahou bolo využiť maximum existujúcej techniky, infraštruktúry a zakúpených licencií na software. Pôvodne sme uvažovali aj o zakúpení komerčných produktov, no postupne sa vykryštalizovala myšlienka postaviť uzol založený čisto na voľne dostupných komponentoch.

Hardware: Organizácia disponovala siet'ovým unicastovým pripojením s rýchlosťou 15Mbit/s. Multicastové pripojenie poskytol CESNET na žiadosť organizácie. K dispozícii bolo niekoľko hardwarových komponentov, z ktorých sa podarilo poskladať tri funkčné počítače. Medzi komponentami sa nachádzali aj dve PCI karty určené na digitalizáciu videa. Dva použité projektory už organizácia vlastnila, rovnako ako zvukový zosilňovač, mixér, mikrofón a kamery.

Additionally purchased hardware:

- Ethernetový prepínač (switch) s podporou protokolu IGMP.
- Systém na potlačenie spätnej väzby – Gentner AP-800.
- Systém reproduktorov.
- Bodové osvetlenie.
- Sekundárna grafická karta do PCI, ktorá umožňuje zapojiť ďalšie dva projektory alebo monitory.
- Polohovateľný stojan na projektor.
- Niekoľko metrov VGA a STP káblových rozvodov.

Software: Po softwarovej stránke nebolo potrebné zakupovať žiadne nové licencie aj napriek tomu, že na jednom z počítačov bol nainštalovaný OS Windows a komerčný balík Microsoft Office. Tieto licencie už organizácia vlastnila. Na ostatných miestach bol použitý operačný systém Linux, ktorý je k dispozícii zdarma.

Deployment diagram: Obrázok 5.1 znázorňuje schému zapojenia všetkých hardwarových komponentov. Zatiaľ systému je rozložená na tri počítače podľa ich zamierania – **zobrazovanie, kódovanie a premostenie**.

5.4.1 AG-Display

Jeho hlavnou úlohou je zobrazovať všetkých účastníkov konferencie pomocou projektorov na projekčné plátno. Aby sme získali dostatočný počet VGA výstupov, nainštalovali sme dve grafické karty (dual-head). Jednu z kariet sme navyše prinútili klonovať obraz z VGA výstupu na S-Video výstup, čo umožnilo nahrávanie signálu na mini-DV kazetu.

Operačný systém Windows bolo nutné použiť z dôvodu zdieľaných prezentácií z aplikácie PowerPoint. V dnešnej dobe už však existuje tzv. uno-driver pre OpenOffice.org, ktorý poskytuje rovnakú funkčnosť. Druhým dôvodom použitia Windows je software určený na ovládanie systému Gentner AP-800. Dnes už máme možnosť emulovať túto aplikáciu v prostredí Wine.

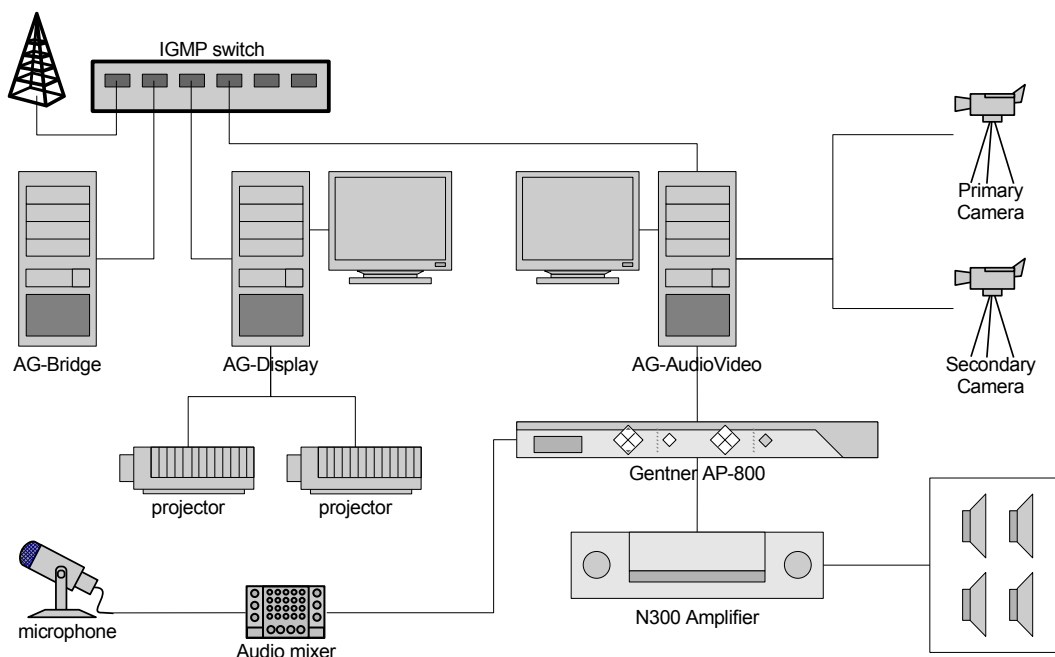


Figure 5.1: Architektúra Ciant AG node.

AG-Display comprises of following components

- CPU Intel Celeron 1.4GHz, RAM 512MiB, 100Mbit Ethernet.
- Windows 2000 Professional operating system.
- Direct connection to IPv4 multicast-capable network.
- Three VGA outputs to external beamers, single VGA output to LCD monitor.
- AccessGrid version 2.4.
- MS Office suite.
- G-Ware: Gentner AP-800 controlling software.

5.4.2 AG-AudioVideo

Jadrom nášho videokonferenčného systému je počítač s operačným systémom Linux, ktorý obsahuje dve PCI karty určené na digitalizáciu obrazu pomocou S-Video vstupu. Do týchto kariet sú zapojené primárna a sekundárna kamera (obrázok 5.1). V našej implementácii sme nepostupovali presne podľa doporučení uvedených v dokumente Access Grid Hardware Specification [38], ale rozhodli sme sa spojiť do jedného hardwarového celku komponenty Video Capture Computer, Audio Capture Computer a Control Computer. Výkon procesoru v našom systéme bol zatiaľ postačujúci aj pre desiatky účastníkov.

AG-AudioVideo comprises of following components

- CPU Intel Celeron 1.4GHz, RAM 512MiB, 100Mbit Ethernet, SB Live.
- Operačný systém Linux Suse 9.3.
- Priame pripojenie na IPv4 sieť s podporou multicastu.



Figure 5.2: Gentner AP-800 echo cancellation device.

- VGA výstup na LCD monitor.
- AccessGrid verzia 2.4.

5.4.3 AG-Bridge

Problémy s pripojením klientov bez multicastovej siete rieši server AG-Bridge. Premostenie unicast-multicast je zabezpečené pomocou komponentu QuickBridge. Stále však ostáva problém s doručovaním UDP paketov ku klientom za firewallom.

AG-Bridge comprises of following components

- CPU Intel Celeron 300MHz, RAM 255MiB, 100Mbit Ethernet.
- Operačný systém Debian GNU/Linux.
- Priame pripojenie na IPv4 sieť s podporou multicastu.
- AccessGrid verzia 2.4, z toho používame iba VenueServer a QuickBridge.

5.4.4 Gentner AP-800

Obrázok 5.2 ilustruje zariadenie špeciálne navrhnuté pre potlačenie spätnej väzby, ktorá prirodzene vzniká pri snímaní zvuku citlivým mikrofónom v miestnosti ozvučenej sústavou reproduktorov. Matica vstupov a výstupov 12x12 je programovateľná pomocou aplikácie G-Ware. Obvykle sa cena tohto zariadenia pohybuje v prepočte okolo 75000 CZK. Ponuky na eBay sa pohybujú v prepočte okolo 12000 CZK. Existujú aj lacnejšie verzie napr. AP-400.

5.5 Videoconferencing room

Miestnosť, ktorú sme sa rozhodli využiť pre účely AccessGridu, slúžila pôvodne ako kancelária organizácie. Pri návrhu sme preto mali stanovené obmedzujúce podmienky, pretože sme museli zachovať jej pôvodnú funkciu. Spôsob, ktorým sme sa vysporiadali s týmto problémom, ukazuje, že je možné aj v iných organizáciách postupovať podobne s minimálnymi nákladmi. Hlavné zmeny:

- Príprava projekčných plôch, kde sme využili dve steny, ktoré sme pred použitím museli premaľovať novou vrstvou bielej farby.
- Inštalovali sme reproduktory v rohoch miestnosti.
- Inštalovali sme dodatočné bodové osvetlenie v rohu miestnosti.
- Pripravili sme sústavu káblových rozvodov, ktoré pozostávajú z VGA (pc–projektor), S-VIDEO (kamera–pc), STP (pc–switch) a audio (mikrofón–mixer, zosilňovač–reproduktory) rozvodov.

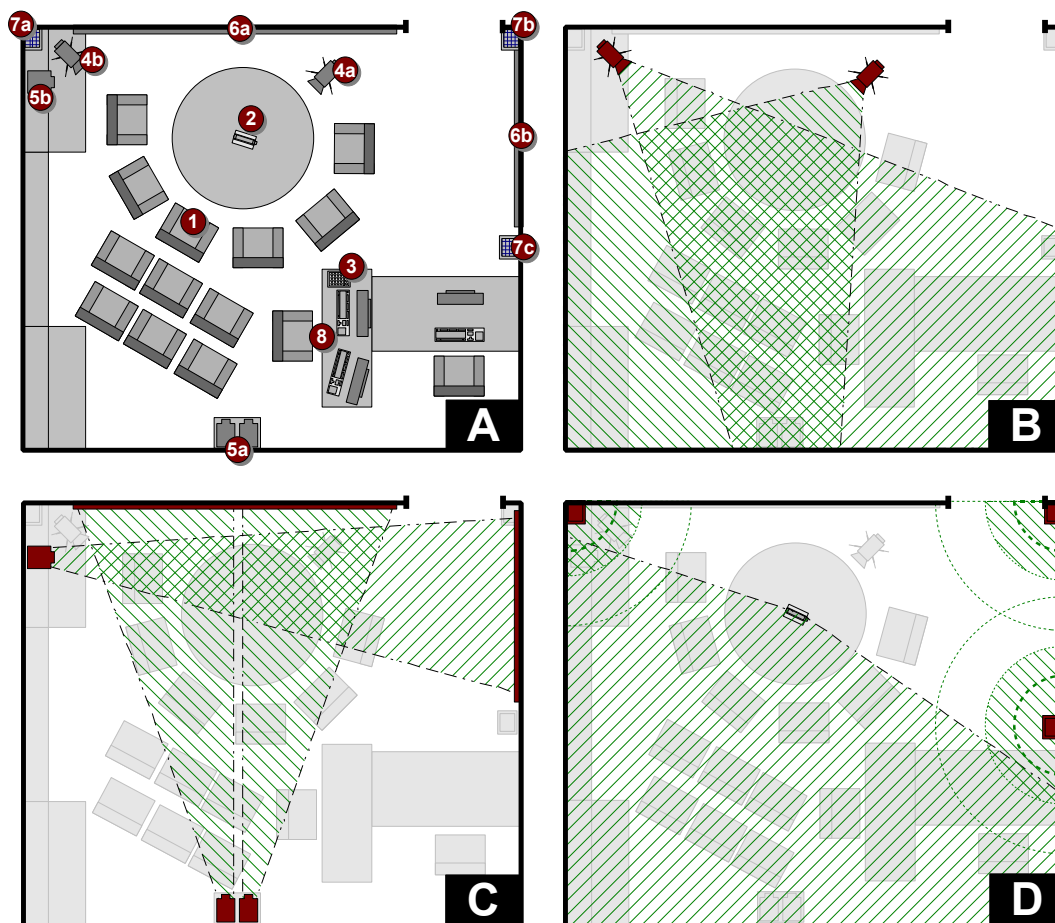


Figure 5.3: Miestnosť určená pre videokonferenčné účely.

Obrázok 5.3(A) popisuje rozloženie objektov v miestnosti. Pre účely video-projekcie sme použili dve biele steny (6a, 6b). Projekciu zabezpečujú 3 projektory (5a, 5b), čo je znázornené na obrázku 5.3(C), pripojené k počítaču AG-Display (8). Mikrofón (2) je umiestnený v prednej časti miestnosti. Umožňuje snímať zvuk pod dostatočne veľkým uhlom, čo je zázorné aj na obrázku 5.3(D) spolu s rozložením reproduktorov (7a, 7b, 7c) v rohoch miestnosti. Zvuk z mikrofónu je prenášaný cez audio-mixér (3) priamo do počítača AG-AudioVideo (8). Dôležitou podmienkou je správne fungovanie zariadenia na potlačenie ozveny (Gentner AP-800). Z obrázku 5.3(D) je zrejmé, že mikrofón bude snímať veľkú časť zvuku pochádzajúceho z reproduktorov, čo by viedlo ku vzniku nepríjemnej ozveny vo všetkých pripojených uzloch.

Vzhľadom na malé rozmery nemôže v miestnosti sedieť viac ako 15 ľudí, čo sa zatiaľ neukázalo ako problém, pretože sme tento počet ešte nedosiahli. Účastníci sú rozdelení do dvoch kategórií – aktívni a pasívni. V prednej časti miestnosti, okolo mikrofónu (2), sedia aktívni účastníci. Na mieste označenom (1) sedí moderátor. Tým je určené aj rozloženie kamier (4a, 4b) znázornené na obrázku 5.3(B). Hlavná kamera (4a) sníma postavu moderátora a jeho najbližších susedov. Vedľajšia kamera (4b) sníma väčšinu publika. Operátor sedí na mieste (8) a má na starosti technické

stránku konferencie. V prípade problémov komunikuje s ostatnými operátormi v textovej forme.

5.6 Day-to-day operation experience

Pôvodne sme počítali s využitím troch projektorov, kde dva hlavné mali slúžiť pre zobrazovanie obrazu z aplikácie VIC a tretí mal slúžiť pre zobrazovanie ovládacích prvkov počítača AG-Display a zobrazovanie zdieľanej prezentácie. V priebehu niekoľkých konferencií sme však zistili, že vystačíme s dvoma projektormi. Navyše spôsob nahrávania konferencie, ktorý sme vtedy používali umožňoval zaznamenať obraz iba jedného projektora.

Prvé pokusy o nahrávanie konferencie sa aj napriek svojej jednoduchosti a problémom ukázali veľmi použiteľné. Výsledky si záujemcovia môžu prezrieť na wiki-stránkach projektu [68], konkrétne záznamy z už spomínaných konferencií Polylogues [32], [31] a [33].

Nahrávanie prebiehalo tak, že sme využili schopnosť grafickej karty klonovať obraz jedného VGA výstupu na S-Video výstup. Tento signál sme preniesli do digitálnej kamery a nahrali na Mini-DV kazety. Po skončení konferencie sme záznam zdigitalizovali do výsledného FLV prípadne MPEG súboru.

Výhodou tohto riešenia vidíme hlavne v jeho jednoduchosti a v tom, že žiadnym spôsobom nezasahuje do konferencie po softwarovej stránke. Nemusíme spracovávať dátový tok RTP paketov a odpadajú problémy so zvyšovaním latencie a jitter. Výsledkom je Mini-DV kazeta, ktorá zároveň slúži ako vhodné zálohovacie médium.

Nevýhod je však hneď niekoľko. Obraz z grafickej karty vo forme S-Video signálu neposkytuje rozlíšenie viac ako 1024x768. Už toto rozlíšenie je hardwarovo preškálované na 800x600, takže nahrávaný obraz má o niečo nižšiu kvalitu ako pôvodný obraz zobrazovaný na projekčné plátno. Obraz, s ktorým teda môžeme disponovať nevyužíva možnosti technológie AccessGrid. Navyše môžeme takto nahrávať iba obraz z jedného projektora. Toto obmedzenie sme riešili tak, že sme ďalšie projektory jednoducho nepoužívali, prípadne sme na ne umiestňovali menej dôležitý obsah. V tom prípade musí operátor konferencie správne rozhodovať o dôležitosti jednotlivých video signálov a zdieľaných prezentácií. Pre operátora to znamená zvýšenú psychickú záťaž, pretože okrem technického zabezpečenia priebehu musí venovať svoju pozornosť aj obsahu konferencie. Obzvlášť náročné to môže byť v prípade, keď sa konferencia zaoberá pre operátora príliš vzdialenou problematikou.

Ďalšou nevýhodou je nutnosť dodatočnej konverzie Mini-DV formátu do formátu MPEG. Toto riešenie navyše neumožňuje priame prehrávanie uložených dátových tokov v novej konferencii. Všetky zložky sú zmixované v jednom obrazovo-zvukovom zázname tak, ako ich účastníci videli pomocou projekcie.

Už sme spomenuli, že kapacita našej miestnosti je 15 ľudí. V takto preplnenom priestore je nutné zabezpečiť dobré odvetrávanie, čo ale naše riešenie v podstate ignoruje. Odvetrávanie je riešené jednoduchým spôsobom – otvorením okna. Už pri našich prvých prenosoch sa však objavil problém, keď zvuk a svetlo mimo miestnosť rušilo citlivé podmienky vo vnútri miestnosti. Otvorenie okna tak nepripadalo do úvahy a účastníci museli vydržať neprijemné teplo a vydýchaný vzduch.

Dôležitým faktorom je osvetlenie miestnosti. Na jednej strane požadujeme tmavé prostredie pre správne fungovanie projekcie. Na strane druhej potrebujeme snímať

obraz pomocou kamery, a preto potrebujeme dostatočne osvetliť účastníkov konferencie. Túto ambivalenciu sme vyriešili v prospech osvetlenia, pretože sa ukázalo, že tmavé prostredie nielenže zhoršuje kvalitu snímaného obrazu a uspáva účastníkov, ale dokonca zvyšuje dátový tok obrazu. Kamera totiž pri horších svetelných podmienkach pridáva do obrazu rôzne náhodné artefakty, ktoré zvyšujú náročnosť jeho kódovania.

Pri inštalácií káblových rozvodov sme museli dbať na správne odtienenie jednotlivých prvkov. Stávalo sa totiž, že pri vedení VGA spolu s elektrickými a S-Video káblami vznikalo rušenie, ktoré deformovalo obraz.

□

Chapter 6

NAT/Firewall

Preklad adries (anglicky Network Address Translation) rieši problém s nedostatkom 32 bitových IPv4 adries. Umožňuje totiž opätovne využívať privátne adresy v podsiet'ach, ktoré medzi sebou nekomunikujú. Komunikáciu s okolitým svetom zabezpečuje uzol s dvoma siet'ovými rozhraniami. Prvé z nich má priradenú privátnu IP adresu a druhé verejnú IP adresu. Medzi zariadeniami je spustený preklad adries – NAT. Situáciu znázorňuje obrázok 6.1. Hlavným komponentom pri preklade je **mapovacia tabuľka**, ktorá udržiava informácie o mapovaní adries a portov v paketoch prechádzajúcich medzi privátnou a verejnou siet'ou. Položky v paketoch, ktoré NAT ovplyvňuje, sú:

- zdrojová adresa a port (prechod z privátnej do verejnej siete),
- cieľová adresa a port (prechod z verejnej do privátnej siete),
- kontrolný súčet (pri každom preklade).

V prichádzajúcom pakete z privátnej siete je zdrojová adresa nahradená verejnou adresou brány. Prijemca si tak myslí, že prijíma pakety z brány a generuje odpoveď s cieľovou adresou brány. Pri príchode odpovede sa porovná zdrojová adresa a port a podľa mapovania v tabuľke sa naspäť upraví cieľová adresa a port. Takto upravený paket je doručený uzlu v privátnej sieti. Mechanizmus NAT musí zabezpečiť v oboch prípadoch správne dopočítanie kontrolného súčtu v paketoch.

Základným dokumentom, z ktorého vychádzame, je RFC 3022 [57] (Traditional IP Network Address Translator).

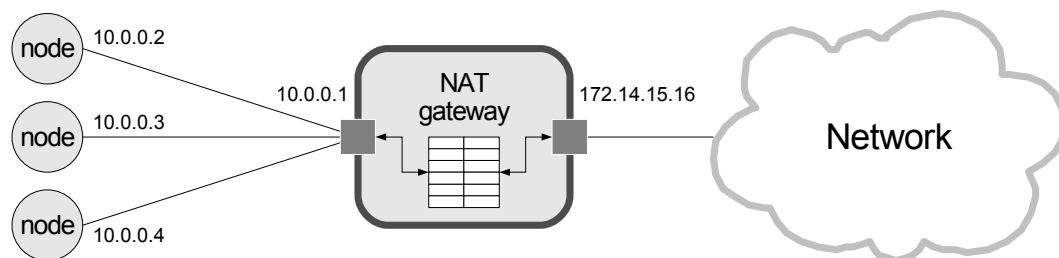


Figure 6.1: Architektúra NAT.

6.1 NAT-related issues

Pri preklade adresy je upravená iba hlavička paketu. Je zrejmé, že tento mechanizmus nespolupracuje s protokolmi, ktoré vkladajú informácie o adresách do tela paketu. Už vôbec nemôže fungovať v prípade šifrovaných paketov.

Mechanizmus NAT umožňuje, aby uzly z privátnej siete kontaktovali uzly s verejnou IP adresou. Problém však nastane v opačnom smere. Táto nevýhoda môže slúžiť ako istá forma bezpečnostného prvku, ale znepríjemňuje komunikáciu VoIP technológiám. V našom prípade sa problém prejaví pri premost'ovaní multicastovej siete pomocou nástroja QuickBridge. Pakety smerované uzlom za NATom sú doručené len v prípade, že existuje príslušný záznam v mapovacej tabuľke NATu. Mapovanie je možné vytvoriť jediným paketom vychádzajúcim od uzla z privátnej siete a musí byť udržiavané periodicky vysielanými paketmi (keepalive).

Nepríjemnou vlastnosťou niektorých typov NAT je mapovanie portov v prípade, že viac uzlov z privátnej siete komunikuje s rovnakým verejným uzlom na rovnakom porte.

6.2 A simple NAT-traversal test

Prechod UDP paketov cez NAT je možné v praxi overiť nasledujúcim pokusom. Testovacia konfigurácia obsahuje komponenty: gateway, server a klient.

NAT gateway: Počítač s operačným systémom Linux, ktorý disponuje dvomi sieťovými kartami – `$NAT_PUB_DEV` s adresou `$NAT_PUB_IP` smerujúcou do verejnej siete a `$NAT_PRIV_DEV` s adresou `$NAT_PRIV_IP` smerujúcou do privátnej siete. Pomocou nástroja `iptables` sme na zariadení `$NAT_PUB_DEV` spustili preklad adresy.

Server with public IP address: Verejná adresa serveru je `$SERVER_IP`. Pre testovacie účely sme použili nástroj `netcat`, ktorý nám umožní vytvoriť UDP server na porte `$SERVER_PORT`. Obsah každého doručeného paketu bude vypísaný na štandardný výstup (stdout) a štandardný vstup (stdin) bude odoslaný naspäť vo forme UDP paketu na rovnakú adresu a port.

Client with private IP address: Umiestnený v privátnej sieti s privátnou adresou `$CLIENT_IP`. Parametre uzla boli nastavené pomocou DHCP, kde hodnota **default gateway** bola nastavená ako `NAT_PRIV_IP`. Znova sme využili nástroj `netcat`. Štandardný vstup (stdin) je odosielaný na adresu servera a prichádzajúce pakety sú vypisované na štandardný výstup (stdout).

Prvý UDP paket odoslaný z uzla s privátnou IP adresou na server s verejnou IP adresou vytvorí cestu pre UDP pakety v oboch smeroch. NAT prekladá pakety zo strany servera na základe vytvoreného mapovania. Táto cesta môže zaniknúť pri reštarte NAT, prípadne po vypršaní časového limitu ak uzly prestanú komunikovať.

```
# gateway: spustenie NAT
iptables -t nat -A POSTROUTING -j MASQUERADE -o $NAT_PUB_DEV
```

```
# server: testovanie dostupnosti
ping $NAT_PUB_IP

# server: listen
netcat -lup $SERVER_PORT

# client: connect
netcat -u $SERVER_IP $SERVER_PORT
```

6.3 Tunneling

Prechádzajúci test ukazuje jednoduchý spôsob prechodu cez NAT v prípade, že komunikácia je inicializovaná z vnútra privátnej siete a ďalej prebieha obojsmerne pomocou jediného portu. Ponúka sa teda možnosť istým spôsobom zabaliť požadované dáta do vnútra jedného dátového toku – **tunelu**. Vytvorený tunel môže poskytovať rôzne služby:

- šifrovanú komunikáciu,
- spoľahlivé doručovanie,
- transparentné tunelovanie,
- tunelovanie multicastových paketov,
- tunelovanie iných protokolov okrem UDP, napr. IPX
- a iné služby ...

Pre multimediálne prenosy je vhodné používať nespoľahlivé doručovanie s minimálnymi nárokmi na balenie a rozbaľovanie obsahu. Šifrovanie môže negatívne ovplyvniť latenciu systému. V prípade AccessGridu je najžiadanejšie tunelovanie multicastových paketov s čo najväčšou transparentnosťou vzhľadom k užívateľovi. Tieto požiadavky spĺňa aplikácia AGConnector, ktorá pre tunelovanie multicastových paketov používa protokol UMTF.

6.3.1 NAT-to-NAT communication

Doteraz sme uvažovali o situácii, kedy klient za NATom komunikoval so serverom s verejnou adresou. Existuje aj možnosť ako môžu spolu komunikovať dva uzly, ktoré sa oba nachádzajú za NATom.

Nasledujúci príklad využíva aplikáciu **nat-traverse** a protokol PPP na vytvorenie tunelu. Riešenie funguje iba za predpokladu, že NAT na oboch stranách nenecháva čísla portov. V príklade má NAT gateway pre klienta adresu `${CLIENT_IP}`, NAT gateway pre server má adresu `${SERVER_IP}`. UDP pakety si oba uzly vymieňajú na porte 40000.

```
#client: vytvori zariadenie ppp0
nat-traverse --cmd= pppd updetach noauth passive notty ipparam \
  vpn 10.0.0.1:10.0.0.2 40000:${CLIENT_IP}:40000

#server: akceptuje PPP spojenie
nat-traverse --cmd= pppd nodetach notty noauth \
  40000:${SERVER_IP}:40000
```


Po vytvorení tunelu je nutné nastaviť smerovacie tabuľky na oboch stranách. Cez vytvorený tunel je možné posielat' pakety z aplikácie QuickBridge, alebo je možné tunelovať multicastové pakety, čo však vyžaduje inštaláciu multicastového smerovača. Nastavenie navyše vyžaduje administrátorské práva aby mohlo vzniknúť PPP sieťové rozhranie (typicky ppp0).

```
#client: smerovanie multicastových paketov
route add -net 224.0.0.0 netmask 255.0.0.0 gw ppp0
route add -net 239.0.0.0 netmask 255.0.0.0 gw ppp0

#server: smerovanie paketov
echo 1 > /proc/sys/net/ipv4/ip_forward
mrouted
```

Aplikáciu nat-traverse je možné využiť okrem PPP aj v spojení s OpenVPN, čo umožňuje zašifrovať celú komunikáciu.

□

Chapter 7

Linux-only AccessGrid

V tejto kapitole popíšeme výstavbu uzla založeného na voľne dostupných komponentoch. Ako základ architektúry použijeme existujúci AG uzol v organizácii Ciant. Naším príspevkom pre AG komunitu je niekoľko aplikácií voľne dostupných na wiki-stránkach projektu [68]. V súvislosti so zdieľanými aplikáciami spomenieme hlavne SharedMPlayer.

Fundamental rules:

- Znížiť množstvo potrebného hardwaru na minimum.
- Sprístupniť pripojenie uzlom za firewallom a NAT.
- Využiť operačný systém Linux a voľne dostupné komponenty.

7.1 Emplacement

Podľa E. Hladkej a P. Holuba [24], je potrebné pred implementáciou videokonferenčného uzla zvážiť jeho priestorovú náročnosť. Uvádza sa [37], že je potrebné vyhraďiť miesto aspoň pre troch účastníkov, počítače a projekčnú techniku. Inak nemá zmysel uvažovať o jeho vytvorení. Musíme však rozlišovať medzi plnohodnotným videokonferenčným uzlom, mobilným videokonferenčným uzlom [41] a tzv. personal interface to grid (PIG), ktorý má zabezpečiť pripojenie pre jednotlivcov bez použitia drahej projekčnej techniky. Rozdiely sú hlavne v množstve potrebného hardware. Zatiaľ čo pri výstavbe plnohodnotného alebo mobilného uzla uvažujeme o použití niekoľkých projektorov, v prípade AG PIG vystačí účastník aj s jedným prenosným počítačom dokonca bez vlastnej kamery a mikrofónu. Má teda na výber, či chce pasívne sledovať vysielanie, alebo prispievať aj vlastnými dátovými tokmi. Ďalej sa uvádza, aké problémy môžu vzniknúť pri komunikácii účastníkov z rôznych časových pásiem. Aby sme zmiernili účinky únavy, musia sa účastníci cítiť v miestnosti pohodlne, čo môžeme ovplyvniť správnym výberom nábytku a osvetlenia.

7.2 Network connectivity

Pre rozhranie PIG (Personal Interface to Grid) stačí použiť sieť s kapacitou okolo 1Mbit/s. Pre plnohodný uzol je táto hodnota niekoľkonásobne vyššia, odporúča

sa aspoň 10Mbit/s. V prípade PIG rozhrania môže účastník využiť unicastové pripojenie cez premostenie (ak existuje), v prípade plnohodnotného uzla sa vyžaduje kvalitné multicastové pripojenie.

7.3 Hardware

Venujme sa teraz hardwaru, ktorý je potrebný na výstavbu plnohodnotného uzla. Hardwarová konfigurácia môže vychádzať z architektúry, ktorú sme popísali v kapitole 5.4. Použijeme teda počítač AG-Display na zobrazovanie pomocou projektorov, AG-AudioVideo na kódovanie obrazu a kódovanie/dekódovanie zvuku a počítač AG-Bridge na premostenie multicastu. Z použitých komponentov je dôležité zakúpiť niekoľko kariet na digitalizáciu obrazu pre počítač AG-AudioVideo a niekoľko grafických kariet pre AG-Display. V prípade, že chceme používať multicastovú sieť, nevyhneme sa nákupu sieťového prepínača¹ (switch) s podporou protokolu IGMP (kapitola 3.1).

V prípade, že použijeme sluchátka s mikrofónom pre každého účastníka konferencie, nemusíme investovať finančné prostriedky do nákupu zariadenia na odstraňovanie spätnej väzby. Musíme si však uvedomiť, že sa tým zníži celkové pohodlie účastníkov. V oboch prípadoch je dôležité dôkladne skontrolovať kvalitu zvuku. Nestačí, že počujeme ostatných účastníkov konferencie. Musíme zabezpečiť, aby aj zvuk z nášho uzla bol v dobrej kvalite.

Čo sa týka projekčnej techniky, odporúčame použiť aspoň dva projektory. Jeden bude slúžiť pre projekciu snímok z prezentácie, druhý bude zobrazovať obrazy účastníkov.

Pred nákupom odporúčame preštudovať stránky accessgrid.org, hlavne databázu hardware [39], ktorú vytvorili členovia AG komunity. Stránky poskytujú prehľad o konkrétnych hardwarových prvkoch, akými sú videokamery, webkamery, mikrofóny, zvukové karty, projektory a mnoho ďalších, vrátane ich parametrov, fotografií a odkazov na stránky výrobcu.

Important notice: Multicastový most musí byť inštalovaný na osobitnom počítači. V prípade, že z jedného socketu čítajú dáta dva procesy, získa tieto dáta posledný proces, ktorý otvoril socket systémovým volaním `bind()`. Riešením by bolo zrkadlenie paketov istou formou proxy servera implementovaného ako user-space proces alebo priamo v jadre operačného systému.

7.4 Operating system and essential software

AccessGrid ako multiplatformná aplikácia existuje v rôznych obmenách pre niekoľko najvýznamnejších operačných systémov vrátane Windows, Linux, FreeBSD, MacOS. Pre nás zaujímavým, je operačný systém Linux. Pri výbere distribúcie hrala hlavnú úlohu dostupnosť inštalačných balíkov. Aj keď oficiálnu podporu má distribúcia

¹Je možné použiť namiesto tohto špecializovaného hardware počítač s niekoľkými sieťovými kartami a s podporou multicast routingu v jadre operačného systému. Z dôvodu latencie, spoľahlivosti a spotreby elektrickej energie však odporúčame použiť hardwarový prepínač.

Fedora, dali sme prednosť OpenSuse 10.0 na základe skúseností s uzlom v organizácii Ciant, kde používame OpenSuse 9.3.

Installed packages:

- AccessGrid-2.4-5
- MPlayer-1.0rc1try2-2.pm.1
- xorg-x11-Xvnc-6.8.2-100
- agvcr-2.1.4-1
- + všetky potrebné závislosti

Druhým použitým systémom je Debian GNU/Linux 3.1. Z pohľadu AG 2.4 nie je táto voľba najšťastnejšia, pretože inštalačné balíky existujú iba pre príbuznú distribúciu Ubuntu. Bolo však potrebné využiť existujúci server, na ktorom, okrem iného beží aj wiki-stránka projektu Ciant AG node. Server je v súčasnej dobe umiestnený na adrese `ciant114.cesnet.cz`.

Installed packages:

- accessgrid2.4-1.4 (z distribúcie Ubuntu)
- + všetky potrebné závislosti

7.5 Video and audio transmission

Jednou z najdôležitejších súčastí AG sú tzv. MBone tools, nástroje pre komunikáciu pomocou multicastovej siete. V rôznych obmenách sa používajú aj v systémoch VRVS a InSORS. Ide o aplikácie určené na prenos obrazu, zvuku a zdieľanie textových dokumentov, kde dátové prenosy sú uskutočňované pomocou multicastového rozšírenia IPv4 protokolu (kapitola 3.1). Rozdiely medzi rôznymi verziami sú hlavne v množstve podporovaných kodekov a vo vzhľade užívateľského rozhrania.

Ak chceme využiť tieto nástroje mimo AccessGrid, sú k dispozícii na adrese ktorú uvádzame v použitej literatúre pod označením [43], pre praktické použitie však odporúčame využiť niektorú s ponúkaných modifikácií. Niekoľko odskúšaných verzií sme záujemcom pripravili na stránkach projektu Ciant AG node [68], ako súčasť tejto diplomovej práce.

7.5.1 OpenMash framework

Open Mash [50] bol navrhnutý pre vedeckú komunitu ako dôsledok potreby práce s multimediálnymi dátovými tokmi v rámci videokonferenčných systémov. Tento balík bol navrhnutý s ohľadom na preklenutie problémov, ktoré sa objavovali v podobných, starších systémoch. Islo hlavne o ich slabú prenositeľnosť do iných operačných systémov, než pre ktoré boli navrhnuté, nedostupnosť zdrojového kódu a neúplná alebo len malá podpora IETF protokolov. Idea vzniku tejto aplikácie pochádza od Larry Rowa a Steve McCanna. Vznik sa datuje do roku 1999. Aplikačný balík OpenMash (re)implementuje projekty ako sdr, nv, vic, vat, wb, mtrace vyvíjaných v komunite MBone.



Figure 7.1: Aplikácie určené na prenos obrazu (vic-ucl, vic-vrvs, vievo).

Po implementačnej stránke je OpenMash riešený ako kombinácia jazyka C/C++ s jazykom TCL/Tk [60]. TCL je jednoduchý ale veľmi silný skriptovací jazyk určený nielen pre prototypovanie aplikácií ale aj pre vývoj skutočne prenositeľných systémov. Jeho multiplatformná grafická nadstavba Tk sa používa aj v iných jazykoch, napríklad Python. OpenMash navyše používa ďalšie rozšírenie – OTCL [63], ktoré pridáva podporu objektov do inak funkcionálneho jazyka. Zatiaľ čo jazyk C++ je použitý pre výpočtovo náročné procedúry, interpretovaný jazyk TCL/Tk je vhodný ako spojovací nástroj a pre interakciu s užívateľom pomocou grafického rozhrania.

OpenMash je dostupný pre všetky hlavné platformy, čo z neho robí ideálny nástroj pre multiplatformné videokonferenčné riešenie. Tejto skutočnosti si boli vedomí aj tvorcovia úspešného videokonferenčného systému VRVS a EVO, ktorý sme popísali v kapitole 2.2.4. Autori použili OpenMash pre aplikácie na prenos obrazu a zvuku.

Jazyk TCL/Tk nám umožnil modifikovať existujúce aplikácie VIC-VRVS a VIEVO z projektu VRVS A EVO a vytvoriť novú aplikáciu EVIC, ktorú poskytujeme záujemcom na stránkach Ciant AG node [68].

7.5.2 VIC

Video Conferencing Tool (VIC) umožňuje prenos obrazu v reálnom čase pomocou protokolu RTP, ktorý uvádzame v kapitole 3.2.1. Prenos zvuku však musí byť riešený inou aplikáciou. Obsahuje podporu pre niekoľko kodekov a ich množstvo a kvalita implementácie sa líši v závislosti na verzii. Asi najstabilnejšími verziami sú VIC-2.8ucl1.1.3 z balíka AG a VIC z projektu OpenMash. VIC umožňuje viacbodovú komunikáciu s využitím multicastovej siete. Špeciálnym prípadom, ktorý využijeme, je možnosť posielat RTP pakety na sieťové loopback zariadenie (localhost). To nám umožní prípravu prezentácie, jej nahrávanie a opätovné zdieľanie v konferencii. Obsah RTP paketov je možné pred odoslaním zakódovať a zvýšiť tak bezpečnosť prenosu. Potenciálny útočník ale stále vidí odosielateľa a príjemcu RTP paketu, ako aj množstvo ďalších „užitočných“ informácií. Implicitným kodekom je H.261. Ako uvádzame v kapitole 4.2.2, tento kodek je až 3-krát efektívnejší ako pôvodný kodek NV. Ďalšími významnými kodekmi sú H.263 a H.263+ (kapitola 4.2.3), ktoré ale v niektorých verziách VICu spôsobujú pád aplikácie. To sa našťastie netýka verzie založenej na OpenMash, ktorý chceme využívať. Existuje aj vývojová verzia VICu s podporou kodeku H.264 (kapitola 4.2.4).

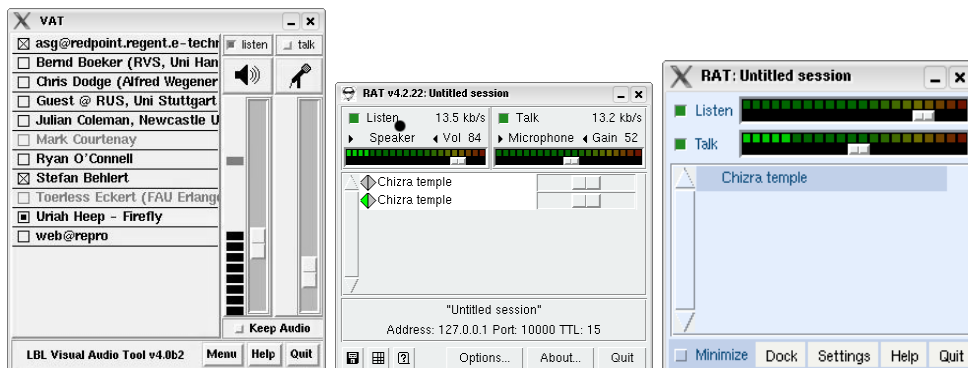


Figure 7.2: Aplikácie určené na prenos zvuku (vat, rat, rat-vrvs).

Na obrázku 7.1 vidíme rôzne verzie aplikácie VIC. Vľavo je originálna verzia VIC-2.8ucl1.1.3, v strede verzia zo systému VRVS založená na upravenom OpenMash 6.1 a vpravo je značne upravená verzia z projektu EVO, ktorá však nepatrí medzi open-source software. Tento problém sa čiastočne pokúšame riešiť reimplementáciou v aplikácii EVIC.

7.5.3 RAT

Robust Audio Tool (RAT) je podobne ako aplikácia VIC určená na prenos multimediálneho obsahu v reálnom čase pomocou RTP paketov medzi účastníkmi multicastovej skupiny. Multimediálnym obsahom je v prípade RATu zvuk. K dispozícii je niekoľko zvukových kodekov, ako napr. GSM, L16 a dokonca v niektorých verziách aj MP3. Líšia sa medzi sebou kvalitou zvuku a spotrebou prenosového pásma. Podobne ako VIC, aj RAT podporuje šifrovanie RTP obsahu a vyskytuje sa v niekoľkých verziách, ktorých príklady vidíme na obrázku 7.2. Vpravo pôvodná aplikácia VAT, ktorá umožňovala prenos zvuku a obrazu súčasne, v strede originálna verzia RAT z balíka AG a vpravo verzia z projektu VRVS.

V kapitole 8.5.2 je popísaná hybridná verzia aplikácie RAT, ktorá vznikla spojením zdrojových kódov z projektu RAT-VRVS a RAT-UCL a umožňuje nahrávať zvuk pomocou sieťového loopback zariadenia (localhost). Takto opravenú verziu ponúkame záujemcom na stránkach Ciant AG node [68].

7.6 Shared applications

Už v základnej inštalácii AG sa nachádza niekoľko zdieľaných aplikácií a navyše je možné túto množinu rozšíriť pomocou jazyka Python. Architektúra zdieľaných aplikácií je zázornená na obrázku 7.3. AG Toolkit poskytuje triedu SharedAppClient, ktorá implementuje všetky operácie potrebné pre komunikáciu pomocou zbernice Event Channel. Trieda uľahčuje doručovanie správ a synchronizáciu klientov. Komunikácia medzi klientami (Application Client) prebieha pomocou XML-RPC.

Ďalším prostriedkom pre vytvorenie zdieľaného prostredia je Virtual Network Computing (VNC) [48]. Systém využíva Remote Frame Buffer protokol (RFB) [47]. Protokol prenáša informácie o zmenách na zdieľanej obrazovke a informá-

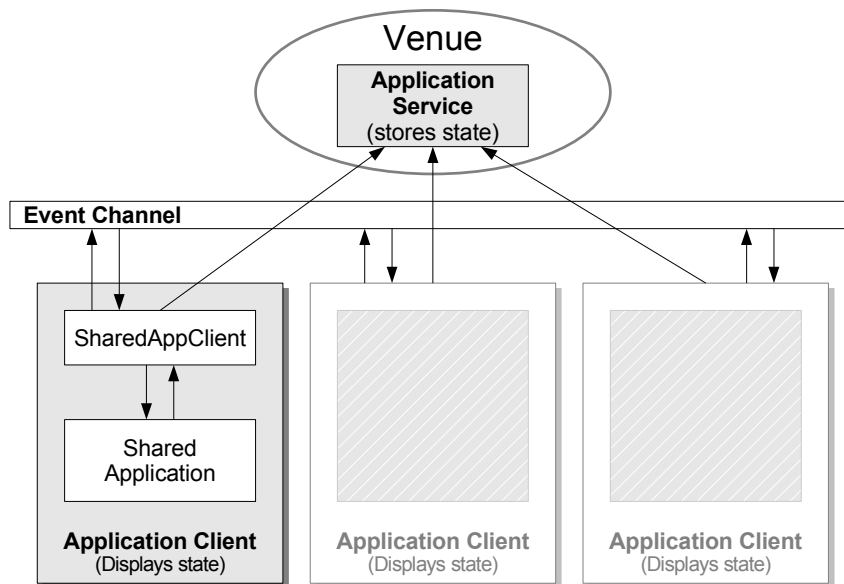


Figure 7.3: Architektúra zdieľaných aplikácií AccessGridu.

ciách o stlačených klávesách a tlačítkach myši. Jeho serverová aj klientská časť je platformovo nezávislá. Protokol RFB je bezstavový a na server môže byť súčasne pripojených niekoľko užívateľov. VNC používa doručovaciu metódu unicast, takže môže vzniknúť problém pri veľkom množstve účastníkov. Projekt VNCast [58] by mal prenášať VNC pomocou multicastových služieb tak, že zakóduje VNC obraz pomocou Motion JPEG (kapitola 4.2.6) do RTP paketov. Zatiaľ je však iba v štádiu vývoja.

Pomocou schémy na obrázku 7.4 chceme poukázať na možnosť využitia VNC ako integračného nástroja formátov, pre ktoré neexistuje špeciálna zdieľaná aplikácia. Uvažujeme o možnosti nahrávania obrazu z VNC vo forme RTP paketov alebo pomocou vncproxy/vncrec/transcode.

```
# Príklad prevodu VNC obrazu do formátu Xvid
vncrec -record file.vnc :0
transcode -x vnc -y xvid -i file.vnc -o file.avi -z -k
```

Komunikáciu protokolom XML-RPC nie je možné nahrávať tak jednoducho ako RTP alebo RFB pakety. Pre účely nahrávania by bolo najvýhodnejšie vysielat' vedľa bežného obrazu aj snímky pracovnej plochy.

Len pre úplnosť uvádzame, že existuje ešte robustnejší protokol pôvodom z Microsoft Terminal Services s názvom **RDP ITU T.128**, ktorý prenáša aj zvuk, súbory a zdieľané disky. V tejto práci však budeme používať jednoduchší systém VNC.

7.6.1 Shared presentations

Linuxová verzia AG 2.4 obsahuje nástroje SharedBrowser, SharedPresentation a SharedPDF. **SharedPresentation** používa OpenOffice pre zobrazovanie prezentácií vo formáte PPT (MS PowerPoint), ODP (Open Document Presentation) a SXI

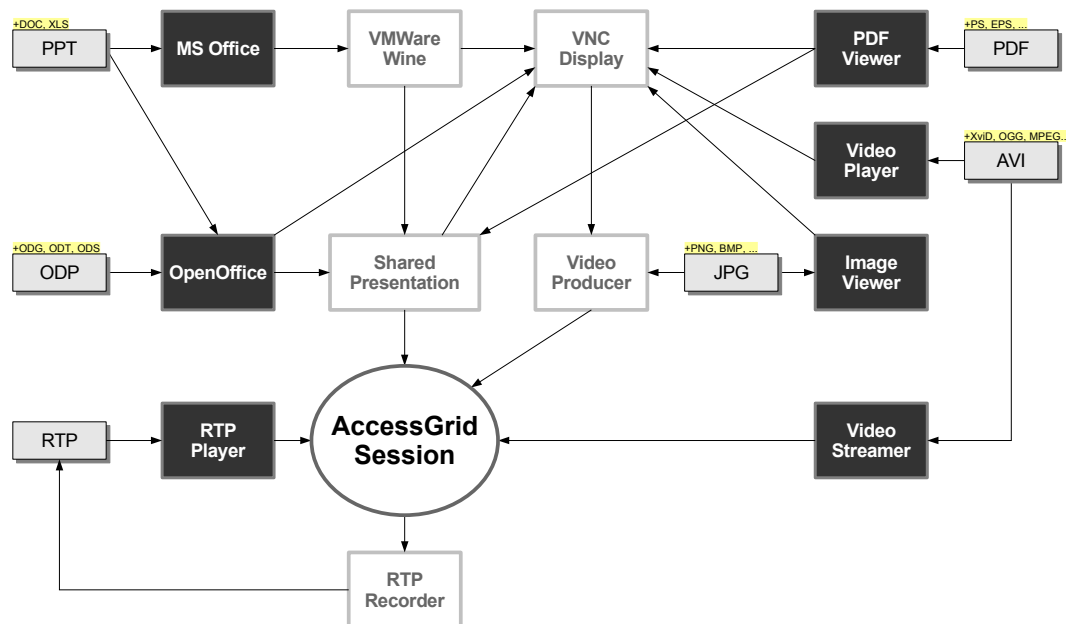


Figure 7.4: Využitie rôznych formátov v prezentácii.

(Staršia verzia OpenOffice/StarOffice). **SharedPDF** umožňuje zdieľať prezentácie uložené vo formáte PDF a pod Linuxom využíva aplikáciu xpdf. **SharedBrowser** využíva internetový prehliadač Mozilla.

Užívateľ iste narazí na niekoľko nepríjemných chýb (alebo vlastností?) týchto aplikácií. V prípade SharedPDF sa zobrazujú vždy dve rovnaké okná s PDF dokumentom, pričom by stačilo len jedno. Využitie by iste našla aj podpora pre prehliadač KPDF. V prípade SharedBrowser musí byť ručne doinštalovaný modul pre Python, ktorý vytvorí rozhranie pre prehliadač Mozilla, inak aplikácia nefunguje. Navyše chýba podpora pre prehliadače Konqueror a Opera. SharedPresentation zas neumožňuje ovládanie fullscreen prezentácie a chýba filter pre súbory s príponou ODP. Opravenú verziu ponúkame záujemcom na stránkach projektu Ciant AG node [68].

7.6.2 Shared desktop

AccessGrid tiež obsahuje dva VNC nástroje – starší VenueVNC a jeho nástupcu SharedDektop [65]. Zdieľanú pracovnú plochu je možné využiť nielen pri prezentáciách ale aj ako nástroj pri riešení technických problémov.

Zaujímavý projekt sľubujú stránky Vislabu, ktoré popisujú aplikáciu Shared-Desktop [40]. Zmenou oproti klasickému VNC je prenos zvuku, 3D akcelerácia a zobrazovanie kurzorov myši všetkých pripojených klientov. Vývoj tejto aplikácie však postupuje veľmi pomaly a zatiaľ neexistuje žiadna použiteľná verzia.

7.6.3 Shared media player

Aplikácia Shared Media Player bola zatiaľ doménou výlučne operačného systému Windows [26]. Rozhodli sme sa preto vytvoriť zdieľanú aplikáciu podobného typu

aj pre systém Linux. Riešenie sa opiera o schopnosti prehrávača MPlayer, ktorý je dostupný pre široké spektrum unixových operačných systémov a dokonca aj pre systém Windows. Pri tvorbe sme vychádzali z aplikácie SharedPDF, ktorá umožňuje zdieľanú prezentáciu PDF dokumentov. Rozhranie k xpdf sme nahradili rozhraním k aplikácii MPlayer.

Inštalčný balík poskytujeme na wiki-stránkach projektu Ciant AG node [67]. Inštalácia vyžaduje spustenie príkazu:

```
agpm.py -z SharedMPlayer.zip
```

7.6.4 Further shared applications

Pre úplnosť by sme ešte radi spomenuli aplikáciu **SharedRasmol** [36], ktorá umožňuje 3D vizualizáciu molekúl, **Shared Question Tool** [1] pre zdieľanú tvorbu otázok a odpovedí a **AGBasicImage**, ktorá umožňuje zdieľané prezeranie obrázkov a úpravu ich metadát. Ďalej **Media Lecture Board**, ktorá umožňuje zdieľanú tvorbu dokumentov. V praxi sa však ukázala ako veľmi nestabilná.

7.7 G-Ware configuration tool

Zariadenia na potlačenie spätnej väzby od firmy ClearOne (predtým Gentner) sa konfigurujú pomocou jednotného nástroja G-Ware. AG uzol v organizácii CIANT využíva verziu AP-800. Ešte donedávna bola situácia komplikovaná tým, že software existuje iba vo verzii pre operačný systém Windows. Aplikácia sa síce používa iba pri výrazných zásahoch do parametrov miestnosti, ale bolo nutné vlastniť aspoň jednu licenciu na inak nevyužívaný operačný systém. Najnovšie verzie Wine a CrossoverOffice však už bezproblémov dokážu tento nástroj emulovať v prostredí Linuxu (obrázok 7.5). Tým skončila posledná závislosť na OS Windows. Inštalčný balík G-Ware s veľkosťou cca 20MiB je možné získať na stránkach výrobcu ClearOne [9] spolu so všetkými potrebnými manuálmi.

Detailný popis inštalácie pre staršie verzie Wine poskytujú stránky Vislabu [59], autori popisujú kompiláciu zo zdrojových kódov vrátane potrebného patch súboru.

7.8 Bridging the multicast

Architektúra AccessGridu je do veľkej miery závislá na multicastovom pripojení. Jeho správne fungovanie sa overuje pomocou nástroja Multicast Beacon. Môže však nastať situácia, kedy niektorý z uzlov dočasne alebo permanentne nepodporuje multicast. Problémy spojené s dostupnosťou muticastu je možné rozdeliť na 3 skupiny:

1. Sieť bez podpory multicastu.
2. Vznik izolovaných multicastových oblastí.
3. Problémy spojené s prechodom cez NAT a firewall.

V histórii AccessGridu sa objavilo niekoľko pokusov o prekonanie týchto problémov. Vznikli tak projekty RCBridge, QuickBridge, AGConnector a komerčný InSORS Enterprise Manager.

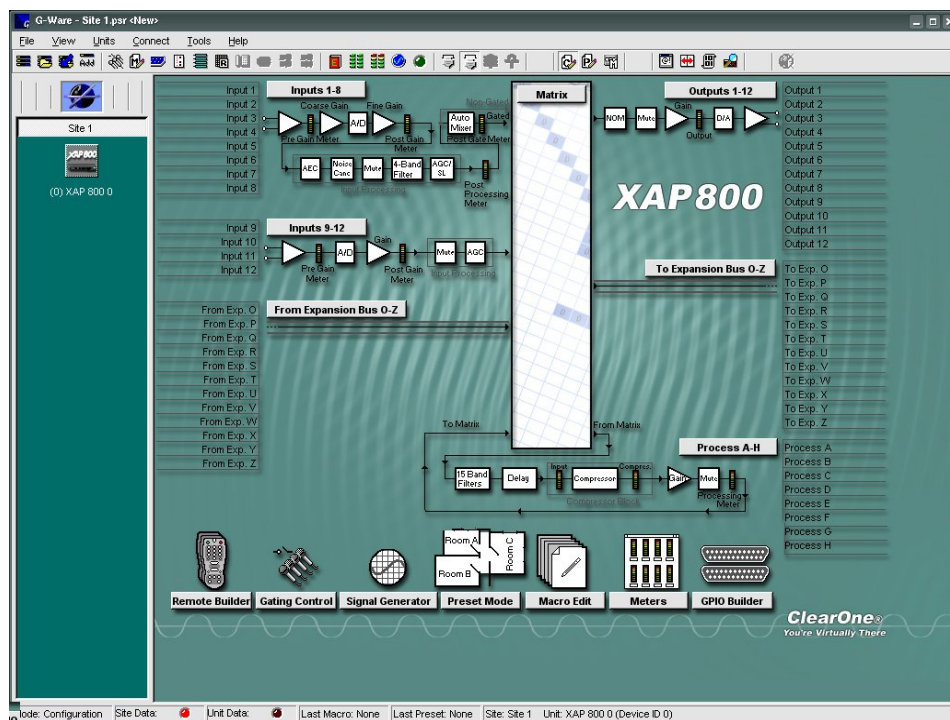


Figure 7.5: G-Ware emulovaný pomocou Wine.

RCBridge: Je nástroj napísaný v jazyku Java s webovým rozhraním, ktorý forwarduje pakety medzi multicastovou a unicastovou sieťou. Nástroj však nie je integrovaný s architektúrou AccessGridu.

QuickBridge: Je jednoduchý UDP forwarder napísaný v jazyku C, ktorý je dodávaný ako súčasť inštaláčného balíka AccessGridu 2.4. Nevýhodou je neúspech pri prekonávaní NAT.

AGConnector: Riešenie založené na myšlienke tunelovania multicastových paketov pomocou jedného UDP spojenia. Obsahuje dva hlavné komponenty – UMTF Agent a UMTF Server. Táto aplikácia existuje len pre AccessGrid 3.0.

InSORS Enterprise Manager: Komerčné riešenie, ktoré okrem multicastového mostu ponúka navyše H.323 bridge a IG Pix Presentation Server pre nahrávanie zdieľaných prezentácií.

7.8.1 QuickBridge

Funkcia komponentu QuickBridge je znázornená na obrázku 7.6. Aplikácia preposiela pakety medzi multicastovou sieťou a unicastovými uzlami použité UDP porty je možné ľubovoľne nastavovať. Detailná dokumentácia, popis inštalácie a zdrojové kódy sa nachádzajú na stránkach projektu AccessGrid [10].

Pre zjednodušenie si predstavme nasledujúcu situáciu: Všetky uzly odosielajú a prijímajú UDP pakety na multicastovej adrese 224.0.0.1 na porte 10000. Jedným

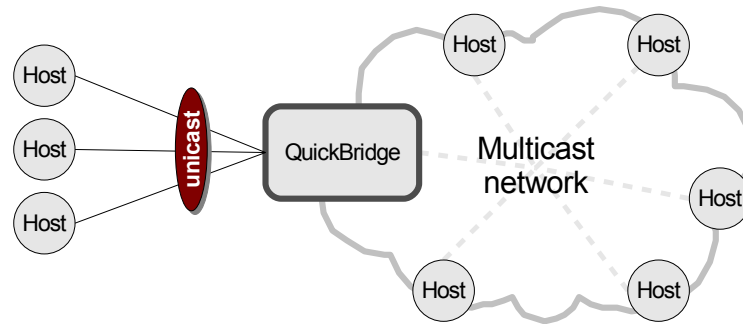


Figure 7.6: Premostenie multicastu pomocou QuickBridge.

z uzlov je aj QuickBridge, ktorý navyše akceptuje unicastové spojenia vo forme UDP paketov na porte 20000. QuickBridge si vedie zoznam aktívnych uzlov, ktoré mu poslali UDP pakety na tento port. Každý UDP paket z unicastovej siete preposiela do multicastovej siete a naopak. Pokiaľ unicastový uzol po istú dobu neodpovedá, QuickBridge ho vyškrtnie zo zoznamu a už mu nepreposiela nové pakety. Takúto jednoduchú situáciu môžeme namodelovať príkazom:

```
QuickBridge -g 224.0.0.1 -m 10000 -u 20000
```

V prostredí AccessGridu sa QuickBridge konfiguruje vždy pre nejaký Venue-Server a to pomocou konfiguračného súboru **BridgeServer.cfg**. Ako príklad konfigurácie môžeme uviesť existujúce nastavenie pre uzol v organizácii CIANT, ktorý sme popisovali v kapitole 5.4.3. Bridge je potom možné spustiť príkazom:

```
BridgeServer.py -c ./BridgeServer.cfg
```

```
# BridgeServer.cfg
[BridgeServer]
name = CiantMagicUnicastBridge
location = Prague
qbexec = /usr/bin/QuickBridge
BRIDGESERVER = CiantMagicUnicastBridge

# Use this port range for the bridge server as a whole
portMin = 20000
portMax = 20005

# Name of Lobby that you wish to provide a bridge to
[https://195.113.187.114:8000/Venues/default]
type = Venue
```

Problém použitia QuickBridge v prostredí s NAT spočíva vo vlastnosti protokolov RTP/RTCP. Pre jedno dátové spojenie sa používajú vždy dva UDP porty – $2k$ pre RTP a $2k+1$ pre RTCP. Cez NAT tak prejde RTCP informácia ale opačným smerom už RTP neprejde.

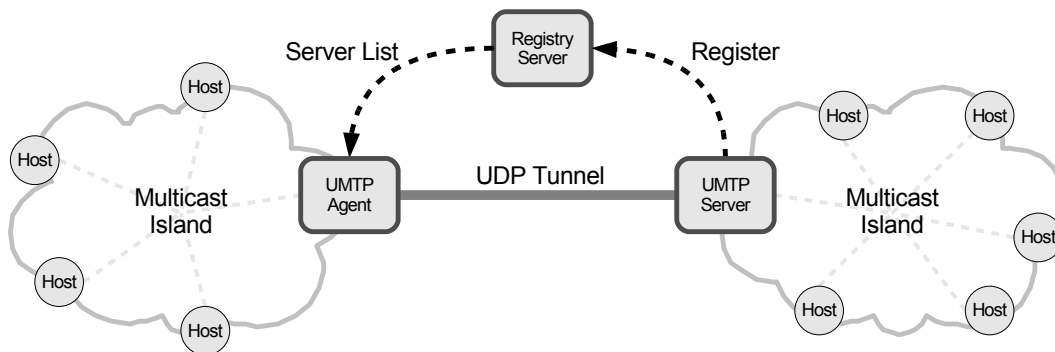


Figure 7.7: Premostenie multicastu pomocou AGConnector. (zdroj [34])

7.8.2 AGConnector

Tento software nahradzuje QuickBridge v novej verzii AccessGridu (verzia 3.0). Rieši problém prenosu multicastových paketov medzi multicastovými ostrovmi, špeciálne prípad siete za firewallom a NATom. AGConnector využíva protokol UMTP 3.1.4, pomocou ktorého tuneluje multicastové UDP pakety medzi multicastovými ostrovmi (obrázok 7.7). Prenos zabezpečuje jediný UDP port 8010 UDP, ktorý nesmie byť blokovaný firewallom. Inštalácia prebieha príkazom:

```
agpm.py -z AG_Connector.zip
```

7.9 Results and suggestions

Všetky komponenty potrebné pre chod plnohodnotného AccessGridového uzla je možné prevádzkovať v operačnom systéme Linux. Problémovými aplikáciami boli G-Ware a SharedMediaPlayer. V rámci tejto práce sme vytvorili zdieľanú aplikáciu SharedMPlayer, ktorá poskytuje podobné funkcie ako SharedMediaPlayer pre Windows. Navyše ťažíme z obrovského množstva kodekov, ktoré podporuje MPlayer. Ostatné zdieľané aplikácie, ktoré sme popisovali, existujú pre obe platformy. G-Ware je možné emulovať v prostredí Wine.

Stav videokonferenčného uzla je potrebné preventívne testovať. Pre tento účel usporiadať komunita AG pravidelné stretnutia zameraná na odhaľovanie a odstraňovanie problémov za pomoci skúsených užívateľov AG. Užitočné informácie o odstraňovaní problémov so zvukom, obrazom, sieťovým pripojením a zdieľanými aplikáciami sa nachádzajú priamo na stránkach AG [3].

□

Chapter 8

Recording the conference

8.1 Recording vs archivation

Na úvod kapitoly zdôrazníme rozdiel medzi pojмами nahrávanie a archivácia. Zatiaľ čo pri nahrávaní je cieľom vytvoriť záznam, ktorý je možné následne zrekonštruovať, pri archivácii sa navyše berie do úvahy celkový kontext, v ktorom bol záznam vytvorený. Na túto skutočnosť by sme radi poukázali hlavne v súvislosti s referenčným modelom OAIS [7], ktorý je ISO štandardom a tvorí framework pre archiváciu s ďalekým časovým horizontom (long-term preservation). Pomerne obsiahly štandard je možné vystihnúť jednoduchou „rekurzívnou“ myšlienkou:

Archivácia dát vyžaduje archiváciu metadáta aj nástrojov, ktoré tieto dáta dokážu interpretovať. Tento postup sa rekurzívne opakuje a zachádza až k archivácii popisu použitých protokolov a kódovaní.

Je zrejmé, že z pohľadu archivácie budú uprednostňované otvorené technológie využívajúce otvorené štandardy, pred uzavretými, nedokumentovanými systémami. Dostávame sa tak až na úroveň zdrojových kódov. Takýto prístup zaručuje, že archivované dáta bude možné spätne rekonštruovať dokonca aj o niekoľko desiat'ročí.

Systém založený na voľne dostupných komponentoch, je teda prvým krokom k úspešnej archivácii. Ostáva už len doriešiť otázku samotného záznamu, definovať, ktoré aspekty komunikácie je vhodné archivovať a podľa toho nájsť vhodné nástroje.

8.2 Data-streams within the AccessGrid session

AccessGrid pozostáva z niekoľkých vzájomne prepojených komponentov. Spôsob komunikácie medzi nimi môžeme rozdeliť na dve hlavné skupiny:

1. prenos multimediálnych dát,
2. prenos riediacich správ.

V kapitole 3.1 sme ako jeden zo spôsobov doručovania popisovali multicast, čo je vhodná prenosová metóda hlavne pre multimediálne dáta pri skupinovej komunikácii. Pre samotný prenos sa využíva protokol RTP.

Na druhej strane, riadiace správy tvoria len malú časť celkovej komunikácie a pre ich prenos sa využíva jednoduchší unicast. Ak hovoríme konkrétne o komunikácii medzi VenueServerom a ostatnými komponentami AG, táto prebieha v istej forme XML-RPC.

8.2.1 Candidates for archivation

Najvhodnejšími kandidátmi pre nahrávanie, archiváciu a spätnú rekonštrukciu budú multimediálne prenosy – zvuk a obraz. Hneď za nimi nasleduje obraz prezentácie a ďalej pracovná plocha. Nevhodným kandidátom sú riadiace správy, či rôzne fragmenty správ týkajúce sa bezpečnosti.

Zvukový a obrazový signál má iste bezstavovú povahu. Mierne komplikácie prinášajú kodeky, ktoré namiesto jednotlivých snímok generujú prechody medzi kľúčovými snímkami. Tento problém sa v najhoršom prípade rieši preskočením k nasledujúcemu kľúčovému snímku. To platí aj pre fragmentáciu dát v jednotlivých ISO/OSI vrstvách. Chceme tým naznačiť, že obraz a zvuk vo forme RTP paketov bude pomerne jednoduché archivovať. V prípade rekonštrukcie poškodeného záznamu bude najkomplikovanejšou časťou rekonštrukcie nájsť nasledujúci kľúčový snímok, čo závisí na použítom kodeku.

Ďalším kandidátom je protokol RFB používaný v komunikácii medzi VNC serverom a klientom. Protokol je bezstavový a konzistenciu VNC session nepoškodí ani ukončenie, či výpadky spojenia.

Už sme spomenuli, že AG používa pre ovládanie svojich komponentov protokol XML-RPC. V podstate nás táto komunikácia nemusí zaujímať až na jednu výnimku – pomocou XML-RPC sa totiž synchronizujú zdieľané aplikácie AccessGridu. Hlavne zdieľaná prezentácia je horúci kandidát na nahrávanie. Jedným z možných riešení by bolo ukladať udalosti prechodov medzi snímkami pomocou špeciálne upravenej verzie SharedPresentation. Problémom je však tvorba univerzálneho nahrávacieho nástroja pre všetky zdieľané aplikácie. Navyše takto uložené informácie je možné spätne prehrávať len rovnakým spôsobom, akým boli vytvorené. V kontexte OAIS by sme museli archivovať aj všetky použité prehliadače a zdieľané aplikácie.

8.2.2 Conversion to common format

...or how to avoid n^2 conversion tools while using solely n formats

Aby sme proces archivácie čo najviac zjednodušili, rozhodli sme sa všetky udalosti, ktoré vzniknú počas videokonferenčného prenosu, previesť do audio-vizuálneho formátu, ktorý uložíme ako RTP pakety. Pri prevode sa kvalita pôvodných dát môže mierne zhoršiť, no získame jednotný formát, pre ktorý môžeme neskôr vytvoriť nástroje na prehrávanie a úpravu.

VNC display: Prevod vyžaduje snímanie obrazu pracovnej plochy a kódovanie pomocou vhodného kodeku do formátu RTP.

Zdieľaná prezentácia: Stačí ak budeme snímať tú časť pracovnej plochy, ktorá obsahuje zobrazenú prezentáciu. Vhodne uplatníme prechádzajúci prípad a využi-

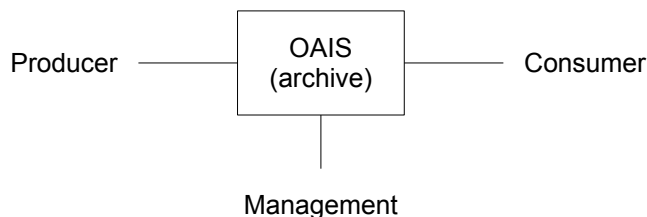


Figure 8.1: Prostredie pre archiváciu podľa OAIS[7].

jeme VNC pre zobrazovanie prezentácie. Toto riešenie sa neobmedzuje iba na zdieľanie prezentácií vo formáte PDF, či ODP.

Referenčný model OAIS definuje 4 prvky archivačného prostredia (obrázok 8.1). Z nášho pohľadu bude „Producer“ každá aplikácia, ktorá vysiela RTP pakety, teda VIC a RAT. Prvok „Consumer“ bude užívateľ, ktorý sa snaží archivovaný záznam znova rekonštruovať a prehrať napr. pomocou aplikácie VIC a RAT. Ak by sme chceli postupovať podľa referenčného modelu, musíme zabezpečiť aj existenciu entity „Management“, ktorá nastavuje zásady dodržiavané v rámci archivačného prostredia OAIS. O údržbu archívu sa teda nestará management ale štruktúry vo vnútri samotného archívu.

Implementácia modelu OAIS je mimo záber tejto práce a ďalší text je nezávislý na použitej archivačnej paradigme.

8.3 Existujúce nástroje pre nahrávanie

8.3.1 Tcpdump – Packet sniffer

Tcpdump umožňuje ukladať pakety prechádzajúce sieťovým zariadením. Na prvý pohľad by sa zdalo, že ide o ideálny nástroj na zachytenie kompletnej informácie v dátovom prenose. Problémom záznamu RTP paketov je však ich správne časovanie. Výsledný súbor siete obsahuje všetky audio-vizuálne dáta, ale spätná rekonštrukcia časovania je obtiažna. Tcpdump teda nie je vhodným nástrojom.

8.3.2 RTPtools

Je sada nástrojov pre prácu s RTP dátovým tokom. Nájdeme tu nástroje rtprecord, rtpplay a rtpproxy. Ďalším nástrojom je RTP reflektor, ktorý umožňuje zrkadliť prichádzajúce dáta, dokáže mixovať rôzne RTP toky a transparentne prekódovať komunikáciu pomocou iného kodeku. Všetky nástroje sa spúšťajú z príkazového riadku a poskytujú iba základné funkcie.

8.3.3 AGVCR – RTP recorder/player

Zatiaľ najvhodnejším nástrojom určeným pre nahrávanie RTP paketov je AGVCR od D. Pipera [46], Indiana University. Poskytuje grafické aj textové rozhranie (dokonca pomocou Telnetu), umožňuje automatické spúšťanie aplikácií určených na nahrávanie a prehrávanie multimédií. Výsledkom nahrávania je samostatný súbor, ktorý obsahuje informáciu o všetkých prijatých dátových tokoch a ich časovaní. Ďalej umožňuje export podmnožiny týchto dát rozdelením na menšie časti.

8.4 Nahrávanie prebiehajúcej konferencie

Účelom nahrávania je záznamenat' v čo najvernejšej podobe priebeh konferencie z pohľadu jedného uzla. Najjednoduchšou možnosťou, ktorú sme použili aj v prípade uzla v organizácii Ciant, je nahrávanie signálu z grafickej karty napr. vo forme S-Video signálu. Výhody a nevýhody tohto riešenia sme popisovali v kapitole 5.6.

Druhý komplexnejší prístup využíva práve nástroj AGVCR. Nahrávanie vyžaduje osobitný počítač, pretože nie je možné v jednom operačnom systéme zároveň nahrávať, prezerat' a vysielat' pakety pomocou rovnakého soketu. Vždy posledná aplikácia, ktorá použije volanie `bind()`, bude dostávať pakety, ostatné môžu iba vysielat' pomocou volania `send()` alebo `sendto()`.

V prípade veľkej konferencie je nutné dbat' na dostatočný výkon, diskový priestor a kvalitné sieťové pripojenie takéhoto nahrávacieho uzla. Bude totiž ukladať všetky pakety, ktoré sa v konferencii objavajú. Uzol našt'astie nemusí kódovať ani dekódovať žiadne dátové toky, jedinou úlohou je ukladanie RTP paketov.

8.5 StreamVNC – Nový nástroj pre lokálne nahrávanie

Problém nahrávania konferencie viedol k myšlienke vytvorit' nástroj, ktorý by umožňoval prípravu prezentácie vysielaním na lokálne sieťové zariadenie – localhost. Pri tvorbe systému sme brali do úvahy nasledujúce požiadavky:

- Jednoduchá obsluha.
- Existencia grafického rozhrania.
- Schopnosť ukladať obraz pracovnej plochy, napr. s bežiacou prezentáciou.
- Dostatočná kvalita záznamu.
- Nízke nároky na prenosovú kapacitu.
- Podpora nahrávania zvuku aj obrazu.
- Využitie existujúcich komponentov.
- Jednoduchá úprava výsledného záznamu (strih).
- Možnosť jednoduchého prehrávania.

Prvým problémom, bolo získanie vhodných komponentov, ktoré by spolu dokázali komunikovať a boli by dostatočne konfigurovateľné a ovládateľné. Museli pokryť nasledujúce oblasti:

- Nahrávanie RTP paketov.
- Spracovanie zvuku.
- Spracovanie obrazu z kamery.
- Spracovanie obrazu pracovnej plochy.
- Sprístupnenie vhodných prezentačných nástrojov.

Modelová situácia je znázornená na obrázku 8.2. Kľúčovým prvkom je systém VNC (1), ktorý umožňuje zobrazit' aplikáciu (2) bez interakcie s okolitou pracovnou plochou. Obrázok znázorňuje, že obsah VNC je odosielaný na lokálnu adresu aplikáciou VIC (3) spolu so zvukom z aplikácie RAT (4) a ďalším obrazom z kamery (5). Všetky tri dátové toky sú nahrávané pomocou AGVCR (6).

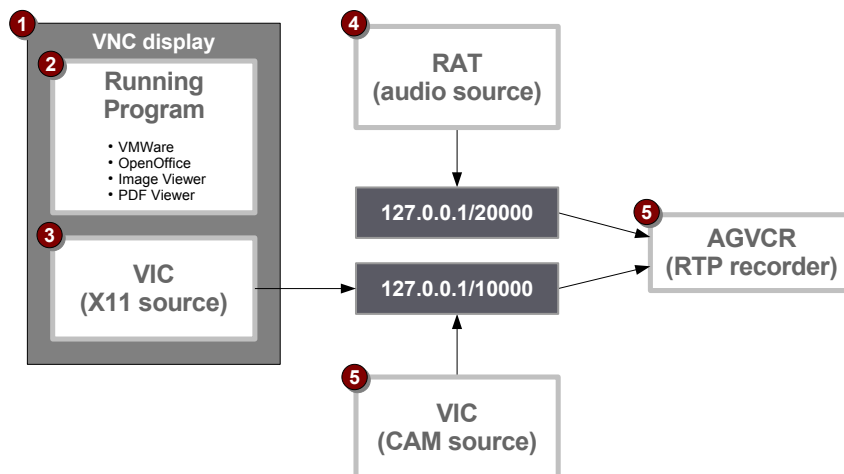


Figure 8.2: Príprava prezentácie pre neskoršie použitie.

8.5.1 Otázka nahrávania RTP paketov

Výborný nástroj AGVCR nám uľahčil situáciu pri výbere vhodného nahrávacieho nástroja. Nástroj nevyžadoval žiadne ďalšie zmeny, pretože vykazuje vysokú stabilitu, výkonnosť a konfigurovateľnosť. Jediný problém vidíme v chýbajúcej funkcii, ktorá by umožňovala vystrihnúť časť uloženého záznamu. V dobe písania textu sme už túto požiadavku adresovali autorovi aplikácie.

8.5.2 Otázka spracovania zvuku

Aplikáciu RAT, ktorá umožňuje prenos zvuku pomocou RTP paketov sme museli pred použitím mierne upraviť. Problém štandardne dodávanej verzie RAT 4.3 z dielne UCL sa prejavil pri zapnutom firewalli. Aplikácia sa odmietala spustiť, čo bolo pre naše účely neakceptovateľné. Problém spôsobovala komunikácia komponentu „mbus“ použitého v rámci RATu, ktorý sa vždy pri štarte snažil kontaktovať jednu z multicastových adries, čo spôsobovalo zamrznutie v prípade spusteného firewallu. Po čase sa podarilo nájsť inú upravenú verziu RATu z dielne VRVS, ktorá štartovala bez problémov. Nevýhodou bola absencia rozhrania pre zvukový systém ALSA (Advanced Linux Sound Architecture), podporovaným bolo iba rozhranie OSS (Open Sound System), ktoré vykazovalo nízky výkon a problémy pri spolupráci s inými aplikáciami pracujúcimi so zvukom. Použili sme teda zdrojové kódy oboch implementácií a vytvorili sme hybridnú verziu RAT-SVNC, ktorá kombinuje výhody oboch.

8.5.3 Otázka spracovania obrazu pracovnej plochy

Pre spracovanie obrazového signálu je v AG určená aplikácia VIC. Znova sme však nemohli použiť originálnu verziu z dielne UCL. Aplikácia totiž blokuje soket, na ktorý vysiela RTP pakety. Znamená to, že AGVCR nedokáže tieto pakety nahrat'. Navyše obsahuje podporu kodeku H.263 pochybnej kvality, pretože jeho použitie často vedie k chybe „segmentation fault“. Verzia VICu zo systému VRVS, ktorý je založený na projekte OpenMash však obsahovala funkčný H.263 kodek a navyše disponovala

kompaktnejším designom, dôležitým hlavne pre koncového užívateľa. V projekte EVO sme ďalej objavili novšiu verziu VICu s názvom Vievo, ktorej design nás zaujal a rozhodli sme sa použiť jeho základ aj v novom nástroji.

Názov pre novú verziu VICu sme zvolili EVIC. Obsahuje jadro z projektu OpenMash podľa VIC-VRVS a časti designu z Vieva. Do kódu sme pridali ovládacie prvky umožňujúce prepnúť ovládanie nahrávania na externý X11 display. Túto vlastnosť využívame pri nahrávaní obsahu VNC, kedy ovládanie môžeme umiestniť na reálny X11 display.

Výkon VICu z projektu OpenMash nám umožňuje dosiahnuť 2fps pri rozlíšení PAL a 10fps pri rozlíšení CIF. Pre účely prezentácie tieto hodnoty postačujú, navyše sa tým šetrí prenosová kapacita siete. V budúcnosti je však určite potrebné zvýšiť výkon tejto komponenty. Testy ukázali, že pri 8fps v rozlíšení PAL bolo dosiahnuté 100% využitie procesoru Pentium 4 – 1.8GHz.

8.5.4 Otázka spracovania obrazu z kamery

Aplikáciu VIC sme tiež úspešne použili pre snímanie obrazu z kamery pri 25fps pri rozlíšení CIF. Rozlíšenie PAL nie je zatiaľ podporované, obraz z kamery je zamýšľaný ako dodatočná informácia k dôležitejšej prezentácii a zvuku.

8.5.5 Podpora prezentačných nástrojov

Pri voľbe prezentačných nástrojov bol braný ohľad hlavne na podporu PDF, OpenDocument Format (ODF) a PowerPoint (PPT). Ďalej podpora prehliadača obrázkov, prezentácia webových stránok, obmedzená prezentácia video sekvencií a podpora pre virtuálny stroj VMWare. Samozrejmou je jednoduchá rozšíriteľnosť o ďalšie aplikácie. Verzia TightVNC ktorú používame, neobsahuje podporu pre 3D akceleráciu, preto zatiaľ nie sú podporované ani aplikácie využívajúce OpenGL.

Pre VNC display sme ešte okrem samotnej aplikácie potrebovali vhodný window manager. Pri výbere hrali dôležitú úlohu faktory ako veľkosť, náročnosť na systémové prostriedky, a umiestnenie dekorácií okien. Nakoniec sme vybrali FLWM, ktorý umiestňuje dekorácie okien na ich ľavú stranu. Opticky tak nekolidujú s dekoráciami samotného vncviewera. Pred použitím sme FLWM mierne upravili, okrem iného aj pridaním parametra `-e`, ktorý umožňuje po štarte FLWM spustiť ďalší program. Umožňuje nám to napr. nastaviť pozadie obrazovky pomocou príkazu `xsetroot -solid '#004455'`.

8.5.6 Architektúra StreamVNC

Všetky komponenty spomenuté v predchádzajúcom texte boli integrované do jedného systému – StreamVNC – znázorneného na obrázku 8.3.

Pri minimálnych nárokoch na užívateľa je umožnená vysoká miera konfigurovateľnosti. Použitými jazykmi sú C, TCL/Tk a Shell Script. Spustenie vyžaduje bežiaci X server, na ktorý sa umiestnia všetky ovládacie prvky. Hlavnými prvkami architektúry sú:

- nahrávanie RTP paketov,
- snímanie zvuku,

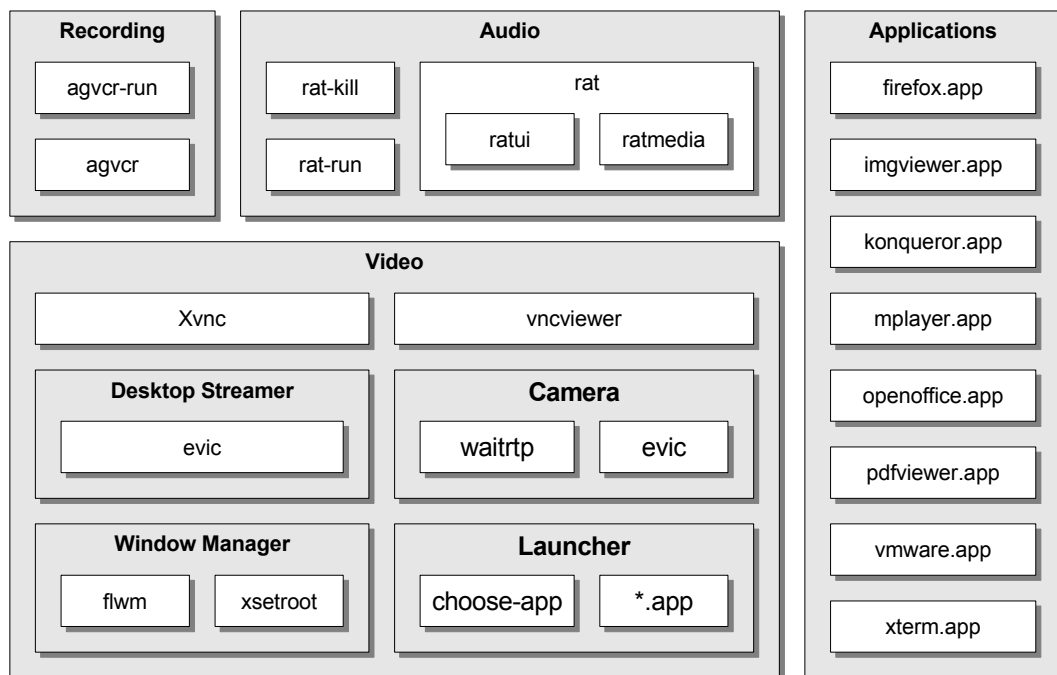


Figure 8.3: Architektúra systému StreamVNC.

- VNC – server, viewer a snímanie pracovnej plochy,
- snímanie obrazu z kamery,
- aplikácie.

Použitie Xvnc (TightVNC server) vyžaduje existenciu fontov `fixed` a `helvetica`. Cesta k fontom obvykle býva správne nastavená v danej distribúcii Linuxu ako je tomu napr. v OpenSuse. Niektoré distribúcie, napr. Gentoo, však vyžadujú explicitné nastavenie `fontpath` pomocou `Xvnc -fp <dir>`. Preto je v rámci StreamVNC poskytnutá binárna podoba Xvnc a najnutnejších fontov spolu s definíciami `fonts.dir` a `fonts.alias`. Server Xvnc sa teda nespúšťa priamo, ale pomocou skriptu `vnc/Xvnc-run`, ktorý nastaví vlastné umiestnenie `fontpath`. V prípade potreby teda stačí zmeniť obsah tohto skriptu.

8.5.7 Implementované profily

Hlavný skript `agvcr-run` sa spúšťa buď bez parametrov, alebo s jedným parametrom reprezentujúcim názov súboru, ktorý sa má prehrať. Aplikácia teda rozlišuje dva profily pomocou dvoch rozdielnych konfiguračných súborov – nahrávanie používa `.agvcr.conf` a reprodukcia používa `.play.conf`.

nahrávanie (record profile): Implicitný profil určený pre nahrávanie prezentácie. Programy znázornené sivou farbou na obrázku 8.4 sa spúšťajú iba v tomto profile. Užívateľ dostane k dispozícii menu, v ktorom zvolí jednu z aplikácií.

reprodukcia (playback profile): V tomto profile začne AGVCR automaticky vysielat' RTP pakety zo súboru, ktorý dostal ako jeden z parametrov. Spúšťa sa iba

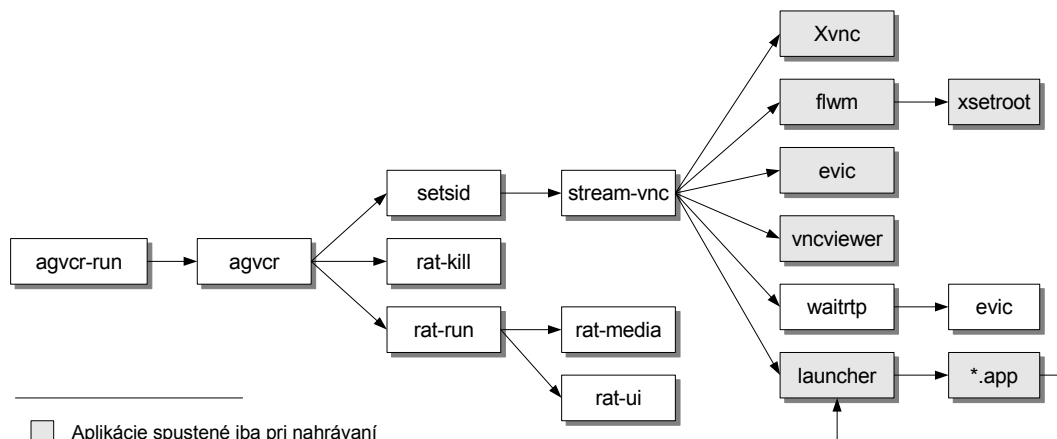


Figure 8.4: Náväznosť programov spustených v rámci StreamVNC.

jedna inštancia aplikácie EVIC po tom, ako prijme jeden RTCP paket. Naopak VNC server ani VNC viewer sa nespúšťajú, pretože pri prehrávaní by zbytočne zaberali priestor na obrazovke. Vysielané RTP pakety sú automaticky prijaté aplikáciou RAT a EVIC a užívateľ vidí obsah uloženej prezentácie.

8.5.8 Správa bežiacich procesov

Kedže sa jedná o integrovaný systém pozostávajúci z množstva komunikujúcich procesov, bola otázka ich správy jednou z najdôležitejších. Hlavný spúšťateľný program s názvom `stream-vnc` je vytvorený v Shell skripte. Má za úlohu štartovať synovské procesy, kontrolovať ich činnosť a v prípade potreby ich aj všetky ukončiť. Najväčšou prekážkou bola malá flexibilita tzv. „job control“ v Shell skripte¹, ktorú sme nakoniec vyriešili nasledovne:

Hlavný program si vedie zoznam procesov `$REQ`, ktoré by mali pri skončení spôsobiť ukončenie celého systému. Registruje sighandler pre signál `SIGCHLD`, čo spôsobí, že ak niektorý zo synovských procesov skončí, dostane rodičovský proces od jadra operačného systému automaticky správu (signál 17). Príkazom `jobs -pr` shell zistí zoznam bežiacich synovských procesov. Ak medzi nimi chýba nejaký proces zo zoznamu `$REQ`, spustí sa ukončovacia sekvencia, kedy rodičovský proces rozosiela bežiacim synovským procesom signál `SIGTERM`. Navyše všetky procesy používajúce VNC display sa automaticky ukončia spolu s VNC serverom.

Iný prístup, ktorý sme sa pokúšali implementovať, využíval funkciu terminálu, ktorú poskytuje operačný systém. Terminál je špeciálne zariadenie určené na medziprocesovú komunikáciu. Operačný systém poskytuje systémové volania pre tvorbu a správu tohto zariadenia. Proces, ktorý zariadenie vytvoril, sa stáva vedúcim skupiny procesov, ktoré budú so zariadením komunikovať. Skupina je identifikovaná pomocou skupinového PID, čo je PID jej vedúceho. V prípade, že vedúci ukončí svoju činnosť, operačný systém by mal rozoslať signál `SIGHUP` všetkým procesom v danej skupine. Výhodou je automatické rozosielanie signálov, na rozdiel od nášho aplikačného riešenia v shell skripte. Nevýhodou je, že niektoré verzie

¹Konkrétne išlo o verziu GNU bash 3.00.16(1)-release (i586-suse-linux)

Shell-interpretovej neposkytujú žiadne ekvivalenty systémových volaní `grantpt()`, `unlockpt()` a `ptsname()`, čo musíme riešiť jednoduchým programom v C. Ten otvorí špeciálny súbor `/dev/ptmx`, ktorý vytvorí „master“ – jednu stranu komunikačného kanálu. Unix API obsahuje funkcie `posix_openpt()`, `grantpt(master)`, `unlockpt(master)`, `ptsname(master)`. Posledná spomínaná funkcia vráti názov súboru `/dev/pts/X`, kde `X` predstavuje „slave“ – druhú stranu komunikačného kanálu. Jednoduchý program teda otvorí pseudo terminál, nastaví správne svoje `filedescriptors`, použije funkciu `setsid()` pre vytvorenie novej session skupiny procesov a stane sa vedúcom tejto skupiny. Následne môže zavolať niektorú z foriem funkcie `exec()` a spustiť iný proces vo svojom pamäťovom priestore. Praktické skúsenosti s používaním StreamVNC však ukázali, že mechanizmus, ktorý sme zvolili je postačujúci.

8.5.9 Test konzistencie prostredia

Aby sme uľahčili identifikáciu problémov v prípade chýbajúcich aplikácií, poskytli sme v rámci skriptu `stream-vnc` parameter `--check`. Pred prvým použitím je teda vhodné spustiť `./stream-vnc --check`, ktorý skontroluje konzistenciu systému a identifikuje najbežnejšie problémy. Okrem iného detekuje aj bežiaci `vmmon` modul jadra operačného systému pre podporu VMWare.

□

Chapter 9

Záver

Ciele stanovené v úvode práce sa podarilo úspešne splniť. Našli sme spôsob tvorby videokonferenčného uzla založeného výhradne na voľne dostupných komponentoch. Nástroje, ktoré vznikli v priebehu tejto práce, umožňujú prípravu a nahrávanie konferencie. Spôsob, ktorým sme spojili existujúce komponenty, je v prostredí AccessGridu unikátny.

Za hlavné prínosy považujeme:

1. Vytvorenie videokonferenčného uzla v organizácii Ciant. Ide o druhý uzol AccessGridu v Českej Republike, ktorý môže slúžiť ako model pre uzly v ďalších organizáciách.
2. V rámci tvorby uzla boli vytvorené internetové stránky, ktoré okrem informácií, týkajúcich sa samotného uzla, obsahujú aj kolekciu nástrojov súvisiacich s videokonferenčnými technológiami. Engine použitý pri ich tvorbe je k dispozícii záujemcom.
3. Kontakty získané počas uplynulých aktivít uzla využijeme v budúcej spolupráci medzi MFF UK a MARCEL network, ktorá by mohla vygenerovať ďalšie zaujímavé projekty ako zdroje bakalárskych, magisterských aj dizertačných prác.
4. Praktické skúsenosti získané pri tvorbe uzla umožnili navrhnuť AccessGridový uzol zložený výlučne z voľne dostupných komponentov. Bolo nutné zabezpečiť chýbajúce aplikácie, inak dostupné v komerčných platformách.
5. Vytvorenie zdieľaného multimediálneho prehrávača pre operačný systém Linux – SharedMPlayer, ktorý doteraz v AccessGrid-e chýbal. Za zmienku stojí aj oprava existujúcej verzie SharedPresentation.
6. Ďalší softwarový prvok, ktorý vznikol na základe tejto práce, je EVIC – upravená verzia aplikácie VIC z projektu VRVS. Hlavným cieľom bolo nájsť vhodného kandidáta na nahrávanie obrazovky VNC, čo pôvodný VIC z dielne UCL neumožňoval. EVIC sa stal súčasťou nástroja StreamVNC a poskytuje ovládacie prvky umiestnené na vzdialený X11 display.

7. Pre účely prípravy prezentácií vhodných pre vysielanie do AccessGridovej konferencie vznikol integrovaný nástroj StreamVNC. Umožňuje lokálne nahrávanie obsahu VNC, zvuku a obrazu z kamery. Dátový tok vo forme RTP paketov ukladáme pomocou nástroja AGVCR.
8. Po teoretickej stránke sme popísali hlavné problémy AccessGridu a videokonferenčných technológií vôbec. Na malých príkladoch sme ukázali možnosť prechodu cez NAT a firewall, možnosti tunelovania a premostenia multicastovej siete. Pre uzol v organizácii Ciant sme postavili QuickBridge, ktorý čiastočne rieši problém s nedostupnosťou multicastovej siete. V budúcnosti uvažujeme o nasadení novej verzie AG 3, ktorá obsahuje kvalitnejšie premostenie pomocou UMTP.

Podstatná časť spomínaných výsledkov môže v budúcnosti generovať nové projekty. Videokonferenčné technológie sa neustále vyvíjajú a užívatelia kladú stále nové požiadavky na kvalitu prenosu, funkčnosť a jednoduchosť použitia. AccessGrid potrebuje nové kvalitné zvukové a obrazové kodeky, čo implikuje vylepšenie existujúcich komponentov VIC a RAT. Podľa modelu InSORS a VRVS je vhodné implementovať nové premostenia (SIP, Skype, H.323, Asterisk ...). Taktiež navrhujeme upraviť súčasnú robustnú architektúru pre použitie v stiesnených podmienkach bez negatívneho účinku na kvalitu prenosu v ostatných miestach siete. Použitelnosť sa zvýši zjednodušením rozhrania VenueClient, napr. podľa modelu EVO, VRVS alebo InSORS.

Veríme, že projekt Ciant AG node bude ďalej pokračovať a záujemcov radi privítame na našich wiki-stránkach [68], kde sa nachádzajú najnovšie verzie všetkých spomenutých komponentov a samozrejme aj tento text v elektronickej podobe.

□

List of Tables

3.1	Formát IGMP paketu.	19
3.2	IGMP Membership Query.	20
3.3	IGMP Membership Report.	20
3.4	Formát SAP paketu.	21
3.5	Formát MADCAP paketu.	21
3.6	Formát RTP paketu.	22
3.7	Formát RSVP paketu.	23
4.1	Audio kodeky s latenciou maximálne 50ms. (zdroj [55])	30

List of Figures

1.1	Videoconferencing system 1968. (source [51])	1
2.1	Architektúra systému Skype.	4
2.2	Architektúra Flash Media Server.	5
2.3	Architektúra systému s použitím multicastovej siete.	7
2.4	Monitorovanie distribuovaného systému VRVS reflektorov pomocou aplikácie MonaLisa.	8
2.5	Typická architektúra AccessGridu. (zdroje [24][8])	11
2.6	Dôsledky oneskorenia zvuku podľa ITU-T G.114 [29].	13
2.7	Porovnanie rôznych veľkostí obrazu.	14
3.1	Rozdiely medzi doručovaním metódou unicast a multicast.	18
4.1	Porovnanie kodeku H261 a NV. (zdroj [42])	27
4.2	Porovnanie kodeku H261 a H263. (zdroj [19])	28
4.3	Porovnanie kodeku H263 a H264. (zdroj [30])	28
4.4	Porovnanie kodeku H264 a MPEG-2. (zdroj [30])	29
5.1	Architektúra Ciant AG node.	35
5.2	Gentner AP-800 echo cancellation device.	36
5.3	Miestnosť určená pre videokonferenčné účely.	37
6.1	Architektúra NAT.	40
7.1	Aplikácie určené na prenos obrazu (vic-ucl, vic-vrvs, vievo).	47
7.2	Aplikácie určené na prenos zvuku (vat, rat, rat-vrvs).	48
7.3	Architektúra zdieľaných aplikácií AccessGridu.	49
7.4	Využitie rôznych formátov v prezentácii.	50
7.5	G-Ware emulovaný pomocou Wine.	52
7.6	Premostenie multicastu pomocou QuickBridge.	53
7.7	Premostenie multicastu pomocou AGConnector. (zdroj [34])	54
8.1	Prostredie pre archiváciu podľa OAIS[7].	57
8.2	Príprava prezentácie pre neskoršie použitie.	59
8.3	Architektúra systému StreamVNC.	61
8.4	Náväznosť programov spustených v rámci StreamVNC.	62

Bibliography

- [1] Shared Question Tool, March 2007. http://www.mcs.anl.gov/~lefvert/PROJECTS/ACCESS_GRID/SQT00L/SharedQuestionTool.htm.
- [2] Ch. Allen. The AMF (RTMP discovery), September 2005. <http://www.acmewebworks.com/Downloads/openCS/TheAMF.pdf>.
- [3] J. Bell. Improving your AG in Preparation for a Quality Assurance Test, October 2006. <http://www.accessgrid.org/node/352>.
- [4] L. Berc, W. Fenner, R. Frederick, S. McCanne, and P. Stewart. RTP payload format for JPEG-compressed video. RFC 2435, Network Working Group, October 1998.
- [5] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin and. Resource ReSerVation Protocol (RSVP). RFC 2205, Network Working Group, September 1997.
- [6] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. Internet group management protocol, version 3. RFC 3376, Network Working Group, October 2002.
- [7] CCSDS. Reference Model for an Open Archival Information System (OAIS). Technical Report 650.0-B-1, Consultative Committee for Space Data Systems, January 2002.
- [8] L. Childers, T. Disz, R. Olson, M. Papka, R. Stevens, and T. Udeshi. AccessGrid: Immersive group-to-group collaborative visualization. *Proceedings 4th International Immersive Projection Technology Workshop*, 2000. <http://www-unix.mcs.anl.gov/fl/publications/childers00.pdf>.
- [9] ClearOne. Clearone resource library, April 2007. <http://www.clearone.com/support/library.php?content=main&product=1>.
- [10] M. Daw and J. T. von Hoffman. Guide to Network Bridging on the Access Grid, March 2004. <http://www.accessgrid.org/agdp/guide/network-bridging/1.0/html/book1.html>.
- [11] S. Deering. Host extensions for IP multicasting. RFC 1112, Network Working Group, August 1989.
- [12] B. DuCharme. Moving to openoffice: Batch converting legacy documents, January 2006. <http://www.xml.com/pub/a/2006/01/11/from-microsoft-to-openoffice.html>.

- [13] P. Farkas, M. Pauliny, V. Michalcin, M. Domarack, and M. Kankula. Jednoduchý návod ako používať VRVS videokonferenčný systém, August 2005. <http://vk.upjs.sk/stranky/technology/videostream/VRVS.pdf>.
- [14] R. Finlayson. The UDP multicast tunneling protocol. Internet-draft, Network Working Group, November 2003.
- [15] Open source flash projects, January 2007. http://osflash.org/open_source_flash_projects/.
- [16] FlashMeeting, January 2007. <http://flashmeeting.open.ac.uk/>.
- [17] InstantPresenter.com, January 2007. <http://www.instantpresenter.com/>.
- [18] Red5:open source flash server, January 2007. <http://osflash.org/red5>.
- [19] B. Girod, E. Steinbach, and N. Färber. Comparison of the H.263 and H.261 video compression standards. *SPIE Proc.*, CR60, October 1995. <http://www.lkn.ei.tum.de/~steinb/PUBLICATIONS/spie95.ps.gz>.
- [20] Haivision. HaiVision MPEG-4 AVC performance, July 2006. <http://www.haivision.com/Downloads/HaiVisionMPEG-4AVCPerformance.pdf>.
- [21] M. Handley and V. Jacobson. Session description protocol. RFC 2327, Network Working Group, April 1998.
- [22] M. Handley, C. Perkins, and E. Whelan. Session announcement protocol. RFC 2974, Network Working Group, October 2000.
- [23] G. Herlein, S. Morlat, J. Jean-Marc, R. Hardiman, and P. Kerr. RTP payload format for the Speex codec. Internet-draft, AVT Working Group, October 2005.
- [24] E. Hladká and P. Holub. Komunikace s technologií AccessGrid Point. *Zpravodaj ÚVTMU*, roč. XIII(2):16–20, 2002. <http://www.ics.muni.cz/zpravodaj/articles/266.html>.
- [25] E. Hladká and P. Holub. Videokonference s vysokou kvalitou. *Zpravodaj ÚVTMU*, roč. XVI(3):9–12, 2006. <http://www.ics.muni.cz/zpravodaj/articles/351.html>.
- [26] J. Hoffman and V. Venkataraman. Shared Movie Player, May 2004. <http://www.accessgrid.org/docs/guide/shared-apps/1.1/html/c132.html>.
- [27] ITU. 7 khz audio – coding within 64 kbit/s. ITU-T G.722, International Telecommunication Union, November 1988.
- [28] ITU. Pulse code modulation (PCM) of voice frequencies. ITU-T G.711, International Telecommunication Union, November 1988.
- [29] ITU. One-way transmission time. ITU-T G.114, International Telecommunication Union, May 2003.

- [30] N. Kamaci and Y. Altunbasak. Performance comparison of the emerging H.264 video coding standard with the existing standards. *IEEE Int. Conf. Multi-media and Expo*, July 2003. http://www.ece.gatech.edu/research/labs/MCCL/pubs/dwnlds/h261_analysis.pdf.
- [31] D. Kera, I. Diosi, A. Mazalek, J. Mucska, J. Bolter, V. Sisler, and J. Holder. Polylogues – Game engines/social engines, March 2006. <http://ciant114.cesnet.cz/agnode/index.php/Main/GameEngines-SocialEngines>.
- [32] D. Kera, J. Holder, J. Ippolito, K. Knoespel, H. Redler, L. Kesner, P. Dousa, and Z. Bauerova. Polylogues – Museums of the future, February 2006. <http://ciant114.cesnet.cz/agnode/index.php/Main/MuseumsOfTheFuture>.
- [33] D. Kera, J. Holder, and K. B. Richter. Polylogues – Visualizations in art and science, May 2006. <http://ciant114.cesnet.cz/agnode/index.php/Main/VisualizationsInArtAndScience>.
- [34] N. Kim. AG Connector – a multicast connectivity solution for AccessGrid, December 2006. http://ace.nm.gist.ac.kr/AG_Connector/.
- [35] P. Krčmář. 10 důvodů proč nepoužívat Skype. *Root.cz*, September 2005. <http://www.root.cz/clanky/10-duvodu-proc-nepouzivat-skype/>.
- [36] Argonne National Lab. SharedRasmol, March 2007. <http://www.mcs.anl.gov/~turam/AG/SharedRasmol.html>.
- [37] Argonne National Laboratory. Access Grid node minimum requirements, March 2002. <http://www.accessgrid.org/agdp/guide/min-req/1.0/html/>.
- [38] Argonne National Laboratory. Access Grid hardware specification, April 2005. <http://www.accessgrid.org/agdp/guide/spec/2005-0408/index.html>.
- [39] Argonne National Laboratory. Access Grid hardware, March 2007. <http://www.accessgrid.org/hardware>.
- [40] N. Lowe. VisualisationGrid/SharedDesktop – Vislab Wiki, July 2005. <http://wiki.vislab.usyd.edu.au/moinwiki/VisualisationGrid/SharedDesktop>.
- [41] L. Matyska. Závěrečná zpráva projektu Mobilní uzel AccessGridu, March 2004. <http://www.cesnet.cz/fond-rozvoje/dalsi/pdf/023.pdf>.
- [42] S. McCanne and V. Jacobson. vic: A flexible framework for packet video, November 1995. <http://www-nrg.ee.lbl.gov/vic/rd.html>.
- [43] UCL Network and Multimedia Research Group. Mbone conferencing applications, July 2001. <http://www-mice.cs.ucl.ac.uk/multimedia/software/>.
- [44] S. J. Park. A history of video conferencing (VC) technology, December 2004. <http://myhome.hanafos.com/~soonjp/vchx.html>.
- [45] C. Perkins, I. Kouvelas, O. Hodson, V. Hardman, M. Handley, J.C. Bolot, A. Vega-Garcia, and S. Fosse-Parisis. RTP payload for redundant audio data. RFC 2198, Network Working Group, September 1997.

- [46] D. Piper. AccessGrid Video (Cassette) Recorder – AGVCR, April 2007. <http://iri.informatics.indiana.edu/~dcpiper/agvcr/>.
- [47] T. Richardson. The RFB protocol, October 2006. <http://www.realvnc.com/docs/rfbproto.pdf>.
- [48] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper. Virtual network computing, February 1998. <http://www.cl.cam.ac.uk/research/dtg/attarchive/pub/docs/att/tr.98.1.pdf>.
- [49] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. Session initiation protocol. RFC 3261, Network Working Group, June 2002.
- [50] L. A. Rowe and S. McCanne. OpenMash project – general info, July 2002. <http://www.openmash.org/about/index.html>.
- [51] L. A. Rowe and S. McCanne. Videoconferencing, December 2006. <http://en.wikipedia.org/wiki/Videoconferencing>.
- [52] H. Schulzrinne and S. Casner. RTP profile for audio and video conferences with minimal control. RFC 3551, Network Working Group, July 2003.
- [53] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A transport protocol for real-time applications. RFC 3550, Network Working Group, July 2003.
- [54] H. Schulzrinne, A. Rao, and R. Lanphier. Real time streaming protocol (RTSP). RFC 2326, Network Working Group, April 1998.
- [55] C. Southeren. How to calculate the real bandwidth for an audio codec, May 2004. <http://www.voxgratia.org/docs/codecbw.html>.
- [56] M. Spencer, B. Capouch, E. Guy, F. Miller, and K. Shumard. IAX2: Inter-Asterisk eXchange Version 2. Internet-draft, The Internet Society, October 2006.
- [57] P. Srisuresh and K. Egevang. The IP network address port translator (NAPT). RFC 3022, Network Working Group, January 2001.
- [58] M. Takatsuka, S. Smith, N. Lowe, and Ch. J. Ng. VNCast – Vislab Wiki, October 2006. <http://wiki.vislab.usyd.edu.au/moinwiki/VNCast>.
- [59] M. Takatsuka, S. Smith, N. Lowe, and Ch. J. Ng. Xap400 configuration with linux, October 2006. <http://www.vislab.uq.edu.au/research/accessgrid/software/xap400/>.
- [60] The Tcler’s wiki, 2007. <http://wiki.tcl.tk/>.
- [61] T. Turletti and C. Huitema. RTP payload format for H.261 video streams. RFC 2032, Network Working Group, October 1994.

- [62] S. Wenger, M.M. Hannuksela, T. Stockhammer, M. Westerlund, and D. Singer. RTP payload format for H.264 video streams. RFC 3984, Network Working Group, February 2005.
- [63] D. Wetherall. MIT object Tcl, 1997.
<http://otcl-tclcl.sourceforge.net/otcl/>.
- [64] C. Zhu. RTP payload format for H.263 video streams. RFC 2190, Network Working Group, September 1997.
- [65] T. Zimmerman. Shared Desktop, March 2007. <http://www.westgrid.ca/support/collab/research-agshareddesktop.php>.
- [66] V. Šimko. AGCentral global node listings – Cinat AG node, January 2006.
<http://agcentral.org/Members/vlx/CIANT>.
- [67] V. Šimko. Shared MPlayer for Linux, January 2006. <http://ciant114.cesnet.cz/agnode/pmwiki.php/Thesis/AppSharedMPlayer>.
- [68] V. Šimko and D. Kera. Ciant AccessGrid node, January 2006.
<http://ciant114.cesnet.cz/agnode/>.